Open Research Online

The Open University's repository of research publications
and other research outputs

Recalibrating the SOPHIE Spectroscopic Archive to Identify Mass-Losing Exoplanets

Thesis

How to cite:

McCune, Calum Scott (2016). Recalibrating the SOPHIE Spectroscopic Archive to Identify Mass-Losing Exoplanets. MPhil thesis The Open University.

For guidance on citations see FAQs.

© 2016 The Author

https://creativecommons.org/licenses/by-nc-nd/4.0/

Version: Version of Record

Link(s) to article on publisher's website:
http://dx.doi.org/doi:10.21954/ou.ro.0000ef4e

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data policy on reuse of materials please consult the policies page.

oro.open.ac.uk

The Open University

DEPARTMENT of PHYSICAL SCIENCES

MPhil Report

Recalibrating the SOPHIE spectroscopic archive to identify mass-losing exoplanets

Calum McCune

Academic Session September 2014 - December 2015

Lead Supervisor:	Dr Carole Ann Haswell
Secondary Supervisors:	Dr John Barnes
	Prof Andrew Norton

Department of Physical Sciences

The Open University

Walton Hall

Milton Keynes

MK7 6AA

DATE OF SUBMISSION : 4 FEBRUARY 2016

DATE OF AWARD : 22 APRIL 2016

ProQuest Number: 13834603

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 13834603

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Acknowledgements

I would like to express my gratitude to my supervisor Dr Carole Haswell for familiarising me with all of the background and concepts of this line of investigations. Especially I wish to thank her for her continued support, guidance and patience throughout this last year of research and in the writing of this report. I would also like to thank my second supervisor Prof Andrew Norton for his help and advice during the writing of this report. I also extend thanks Dr John Barnes for his assistance and contributions regarding the programming methodology and software used in this work. Thanks also to my fellow research student Daniel Staab for his help familiarising me with some of the science concepts involved in these investigations. Finally I wish to thank my family for their unwavering support and encouragement throughout.

This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France (Wenger et al. 2000).

The Open University
- 7 JUL 2016
The Library
DONATION
T 523.24 2016
Consultation copy

Abstract

Recent investigations of the WASP-12 system led to the discovery of an extensive thick and stable circumstellar gas cloud within being fed by material from the evaporating hot Jupiter planet WASP-12b (Haswell et al., 2012) (Fossati et al., 2010b). This planetary material veils the disc of the star providing a unique opportunity to probe composition via transmission spectroscopy. This would not be possible except for in this extreme scenario of catastrophic planet disintegration. Prior investigation of this system reported the anomalously low value of $\log R'_{HK}$ = -5.50 (Knutson et al. 2010), which parameterises the activity in the Ca II H& K resonance line cores. A basal value of $\log R'_{HK}$ = -5.1 is expected even for inactive main sequence stars (Henry et al. 1996; Wright 2004). Any stars with values below this lower limit must either be evolved or hosting similar gas shrouds. So far only a few such cases of this phenomenon have been identified but more are expected to be found within existing spectral archives.

The intention of this project was to construct a reliable software pipeline for reducing large numbers of high resolution stellar spectra from the SOPHIE Échelle spectrograph with the eventual aim of producing archival measurements of the stellar activity parameter $\log R'_{HK}$ for many stars. Integral to this goal was developing means of reliably achieving a good background subtraction of Échelle CCD spectra, specifically in the regions of the Ca II resonance lines, vital to the accurate measurement of $\log R'_{HK}$. The background subtraction software SKYBM was developed for this task and was then tested against existing background subtraction methods offered by the Échelle data reduction package ECHOMOP. SKYBM was found to provide the most accurate subtraction of all methods considered. This SKYBM software is therefore deemed

appropriate for future work calibrating archival stellar spectra in terms of log R'_{HK} to aid in the search for stars hosting mass losing planets.

Table of Contents

1. Introduction to Exoplanets

1.1 History and Discovery

Until recently, the only planets we knew about were within our own Solar System. This meant that all theories, models and thoughts about planet origins, and how they came to be in their current arrangement, were influenced just by what could be seen in our own Solar System. It was not known if our Solar System was a unique occurrence, rare or even common. By this point astronomy had already revealed there to be around 100 billion stars in our Galaxy and more than 100 billion galaxies in the known universe, and only one star known to host planets.

This changed in 1995 with the discovery of the first extra-solar planet, a Jupiter-mass planet orbiting a distant solar-type star 51 Pegasi (Mayor & Queloz 1995). This discovery presented a fantastic opportunity for science as now it was known that other planetary systems existed within our own Galaxy. This opened up the flood gates for a plethora of planet discoveries giving birth to the field of exoplanet research and reinvigorating Galactic astronomy.

Since this initial discovery a great number of exoplanets have been found using various detection techniques and today the overwhelming majority of currently known planets are found outside our Solar System. The total number of confirmed planets is now 2038 (exoplanets.eu, Dec 2015), a number that changes on a regular basis, with many additional candidates identified and awaiting validation.

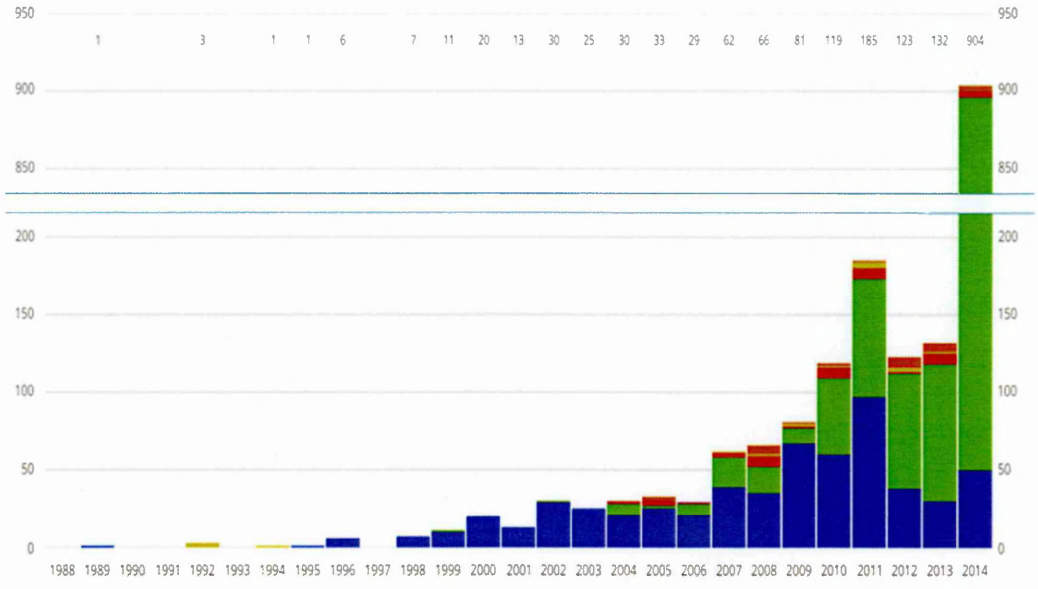


Figure 1: Exoplanet discovery timeline, including detection method. Discoveries made by Direct imaging, Micro-lensing, Transit, timing variations and Radial Velocity methods are shown in red, orange, green, yellow and blue respectively. Image credit: [en.wikipedia.org/wiki/Exoplanet]

Figure 1 above shows how the rate of discovery has accelerated in recent years, made possible as new telescopes have come online as well as advantages in the detection methods covered in Section 1.2. As stated previously, prior to the discovery of extra-solar planets we did not have any other metric to judge whether our Solar System arrangement was typical or unique. In the case of our own Solar System we could only investigate, what our current technology would allow, of the current state and interactions and try to work backwards to identify possible formation histories and migrations. Now with the discovery of a vast population of exoplanets, exhibiting a wide range of characteristics and distributions and at different stages of their respective evolutions, it became possible to discern the underlying processes of planet system formation, the interaction events that can occur and their impact on future dynamics. We could also test previous ideas of how planets formed in general. For example as outlined

by Mordasini et al. (2009), the number of giant planets orbiting inside 1 AU is an important observational parameter for planet formation models.

1.2 Detection Methods

The ever increasing known exoplanet population is credit to various surveys and projects. The types of planets that were discovered were largely dictated by the detection method of these projects each of which have their own particular biases for different planet arrangement. The major detection methods which have been employed for the bulk of discoveries to date are as follows.

1.2.1 Radial Velocity

When a planet is in orbit around a distant star it will exert a gravitational pull on that star inducing a reflex orbit in proportion to the ratio of the two masses and their separation. When observed from Earth (see Figure 2), the system will appear to be in motion with components of velocity in the 'x –y' plane, as projected on the celestial sphere, and in the 'z' direction in our line of sight depending on the inclination of the system. The radial velocity of an object is this motion along the line of sight between two points as the object moves towards or away from an observer. The star's apparent radial velocity will give rise to a Doppler effect in the light we receive. As the star moves away from us, light waves leaving the star are 'stretched' and move towards the red end of the spectrum and likewise as the star moves towards us light waves leaving the star are 'compressed' and move towards the blue end of the spectrum.

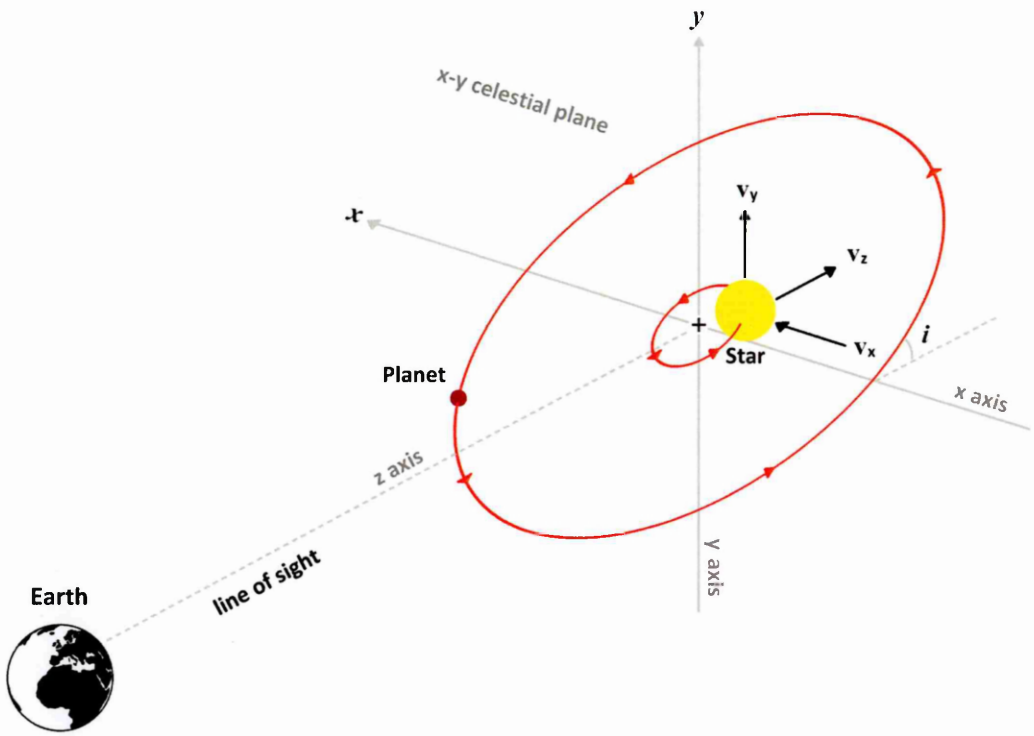


Figure 2: a schematic diagram showing the orientation of the distant stellar system as viewed in Earth's line of sight. Here v_x, v_y and v_z denote the stars components of velocity in the x, y and z directions respectively. The barycentre, +, and orbital inclination, i, of the system are also indicated. The red ellipses trace the orbital paths of the exoplanet and its host star. Only the stars radial velocity component, v_z, brings about the Doppler shift which can be measured by the Earth-bound observer. Stellar motion in the celestial (x-y) plane cannot be measured by this method.

By taking high resolution spectra of the star it is possible to sensitively measure the extent of the respective red shift or blue shift of the light and hence determine the star's radial velocity. Examined over time the apparent radial velocity of the star will change in a regular pattern characteristic of the orbital motion of the system. An example of this radial velocity curve, plotting the star's radial velocity against orbital phase, is shown in Figure 3. Since the extent of the reflex motion of the host star is dictated by the mass of the planet, measuring these variations allows astronomers to deduce the planet's minimum mass, $m \sin i$. This method can only be used to determine a minimum mass, $m \sin i$, as it is only possible to detect the movement of a star towards or away from Earth.

To get a true measurement of a mass, the true motion of the star is required. This can only be determined if the orbital inclination is known.

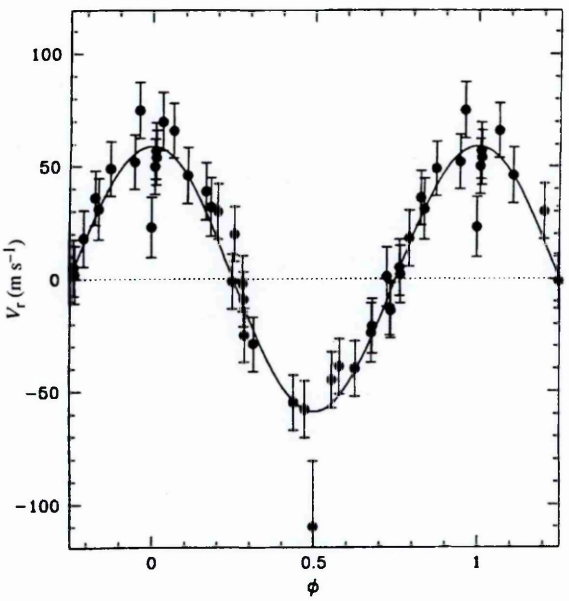


Figure 3: Radial velocity curve for star 51 Peg. Image Source: (M. Mayor & D. Queloz 1995)

In the case where the orbital plane is edge on then the measured radial velocity would represent the true motion of the star and hence the true mass of the planet could be determined. However, if the orbital plane is 'face-on' there will be no radial component of the velocity and therefore no detectable spectral shift. For this reason all radial velocity measurements carry the unknown factor sin *i* unless the orbital inclination, *i*, can be determined using some other method. This presents a particular problem for exoplanet searches as mass is the most important parameter for distinguishing between planets and low mass stars.

This reflex orbit imparted on the host star is described by the Equation [1]. This relation shows that the radial velocity amplitude is highest for massive planets in close-in orbits around low mass stars, creating a bias for these systems using this detection method.

$$A_{RV} = \frac{2\pi a M_P \sin i}{(M_* + M_P) P \sqrt{1-e^2}} \quad [1]$$

In Equation [1] A = amplitude of reflex orbit, a = semi-major axis, M_P = planet mass, M_* = star mass, P = orbital period, e = eccentricity and i = inclination. Additionally for close-in planets having shorter orbital periods less observation time is required to characterise a full orbit. More distant smaller planets induce much smaller reflex orbits taking years making them much harder to detect. Historically this method has been applied to binary star systems, which usually result in RV variations of kms^{-1}. Detecting RV variations caused by companion planets would require instrumentation sensitive to ms^{-1} or even cms^{-1} precision.

At the time when the first exoplanets were being detected the instrumental sensitivity achievable was on the order of several ms^{-1}. In the case of ELODIE which was used to detect the first exoplanet orbiting a main sequence star the sensitivity was 7 ms^{-1}. This would be sufficient to detect Jupiter which exerts a 12.4 ms^{-1} velocity variation on the Sun over a period of 12 years but would not have been sufficient to detect Earth which causes 0.1 ms^{-1} RV on the Sun over one year. As instrumentation has improved, current leading sensitivity in the case of HARPS (3.6 m ESO telescope in La Silla, Chile) has been brought down to < 1 ms^{-1}, with 0.3 ms^{-1} being achievable. This meets the accuracy level required for super-Earth characterisation of <1 ms^{-1} but still not sufficient for smaller rocky Earth like planets.

Future 'third generation' spectrographs are already under development and construction promise sensitivities below 0.1 ms^{-1}. ESPRESSO (Échelle Spectrograph for Rocky Exoplanet- and Stable Spectroscopic Observations) located at VLT is being built with a required specification of 10 cms^{-1}, but aims for a sensitivity of just a few cms^{-1}. CODEX based at the upcoming E-ELT aims for < 2 cms^{-1}.

Radial velocity measurements are the most effective ground based method for detecting exoplanets. This method was responsible for the majority of exoplanet discoveries prior to the launch of the highly successful Kepler space based telescope in 2009. This can be seen in Figure 1 where RV discoveries are shown in blue are dominant until 2012. The inherent advantages and disadvantages of the different detection techniques, which will be discussed, means that collaboration between them can create a more complete picture than any method alone (Bouchy et al. 2010). RV is often used as a follow up method to confirm detections of planets made by other methods.

1.2.2 Transit Method

When orbital alignment allows, distant planets can pass in front of their host stars in our line of sight. This causes a portion of the light from the stellar disc to be obscured resulting in a dip in the total light measured from that star. Generally a giant planet transit will result in a dip of 1 % in the light curve of a host star whereas a terrestrial planet will cause a dip of $\sim 10^{-2}$ %. An example of this dipping in the light curve of a star resulting from a planet's transit can be seen in Figure 4 below. The reliability of transit detections is impeded by the constantly changing transparency in the Earth's atmosphere making the photometric sensitivities required for detecting smaller Earth-like planets impossible for ground based telescopes.

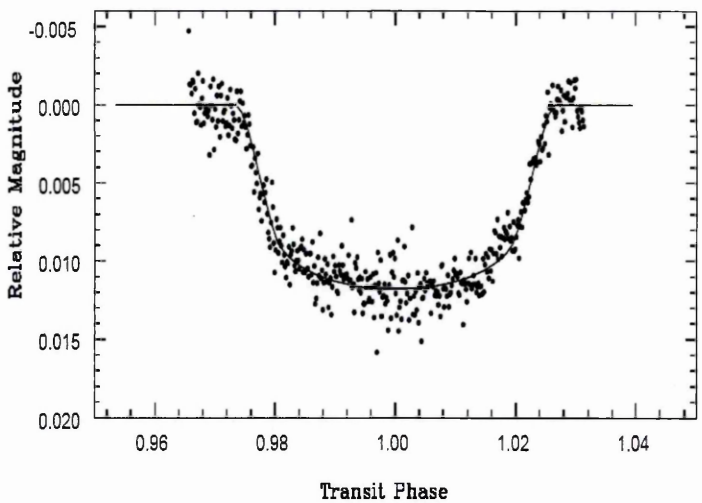


Figure 4: Dip in transit light curve of WASP-14, caused by WASP-14b. Image source: (Joshi et al. 2009)

To achieve the best detection signal from this method a larger dip on the stellar light curve is desired hence this method favours large radius planets in close orbits around dim stars. By characterising the form and extent of the dip in the light curve the planet radius and orbital inclination can be inferred. Since transit detections are the only way to

determine planetary radii and orbital inclinations, planets found using this method are the most informative. Other methods are needed to obtain all other system characteristics and confirm discoveries. Planets discovered using this method are the only planets outside our own Solar System with directly measured sizes. Because of this observations of transiting planets offer the best opportunity to test models of planetary formation.

In most cases a distant planet's orbital plane is neither 'edge-on' ($i = 90°$) nor 'face-on' ($i = 0°$), but instead tilted at some intermediate angle. This imposes the geometric probability, for a random orientation ($0° \leq i \leq 90°$), allowing a planet to be observed transiting the stellar disc of its host star as given in Equation [2].

$$Geometric\ transit\ probability = \frac{R_* + R_P}{a} \approx \frac{R_*}{a} \quad [2]$$

In Equation [2] R_* = star radius, R_P = planet radius and a = semi-major axis. The probability of transit for close-in giant planets is ~10% (for $R_* = 1.16\ R_\odot$, $R_P = 1.2\ R_{Jup}$ and $a = 0.051$ AU) (Seager et al. 2000), meaning since orbital planes are random, only 1 in 10 hot Jupiters will just happen to transit. This probability further depends on the size ratio of the planetary and stellar disc hence increasing with planetary radius and proximity and decreasing with decreasing planetary radius and distance.

The transit technique is extremely useful for characterising the properties of planets, however it is not possible to get a measure of planetary mass using this technique. Additionally similar periodic phenomena can result in similar dips in the light curve, for example eclipsing binaries or starspots traversing the face of the star as it rotates. It is therefore necessary for radial velocity confirmation to identify these false positives

Fortunately, the transit and the radial velocity techniques complement each other well, as the transit technique is capable of determining the orbital inclination *i*, thereby removing the *m* sin *i* uncertainty (as discussed in Section 1.2.1) allowing for accurate measurement of the mass which the transit method alone cannot provide.

To date transit photometry has been by far the most successful method of detection responsible for 1274 of the currently 2038 confirmed exoplanets reported by exoplanets.eu (Dec 2015). This has been primarily due to the success of the space based Kepler telescope which alone has amassed 1030 confirmed planet discoveries with 4696 additional candidates ***[Kepler.nasa.gov Dec-2015]***. Of these 443 are < 2 Earth radii ***[Kepler.nasa.gov Dec-2015]*** including the most Earth-like planet found to date, *Kepler-452b* (Jenkins et al. 2015).

Transmission spectroscopy

In cases where a transiting exoplanet has an atmosphere, light from the background star will not be totally obscured by the outer edge of the planet disc but will instead transmit through. The majority of this light will pass through unaffected, whilst some will be absorbed or scattered by chemical species or clouds in the planet's atmosphere. The absorption of light will occur only at specific wavelengths depending on atmospheric composition. This is because the atmosphere will be optically transparent to the continuum photons but opaque to the photons with wavelengths matching the absorption lines of atmospheric chemical species. The presence of these species will therefore have a proportional absorption effect on specific wavelengths within the stellar spectrum during planetary transit. By carefully characterising the absorption features of the stellar

spectrum both in and out of transit, the absorption due to planetary atmospheric species can be discerned and thus atmospheric composition determined. Uniquely this method gives the opportunity to detect chemical properties of planets offering insight into their formation and evolution.

1.2.3 Direct Imaging

In some cases it is possible to directly image a planet around a nearby host star, however this is notoriously difficult as the host star will likely be on the order of billions of times brighter than the companion planet, for example ~10^9 times brighter in the case of the Sun and Jupiter. For this method to be successful light must be detectable from the planet, either via significant reflection from the illuminated face of the planet (favouring larger radius planets) or by its own thermal emission (as is the case for young planets still shining from Kelvin–Helmholtz contraction). Also this method favours planets in wide orbits, providing sufficient separation from the stellar light source ensuring the two light sources can be spatially resolved. It is also required that the system is relatively nearby so that the angular separation between the host star and the planet can be accurately resolved.

Observations of this kind are usually done in the infrared as here the ratio of spectral energy between the emitted starlight and the reflected light from the planet is at its smallest, making 'planet-shine' easier to discern from the overwhelming starlight.

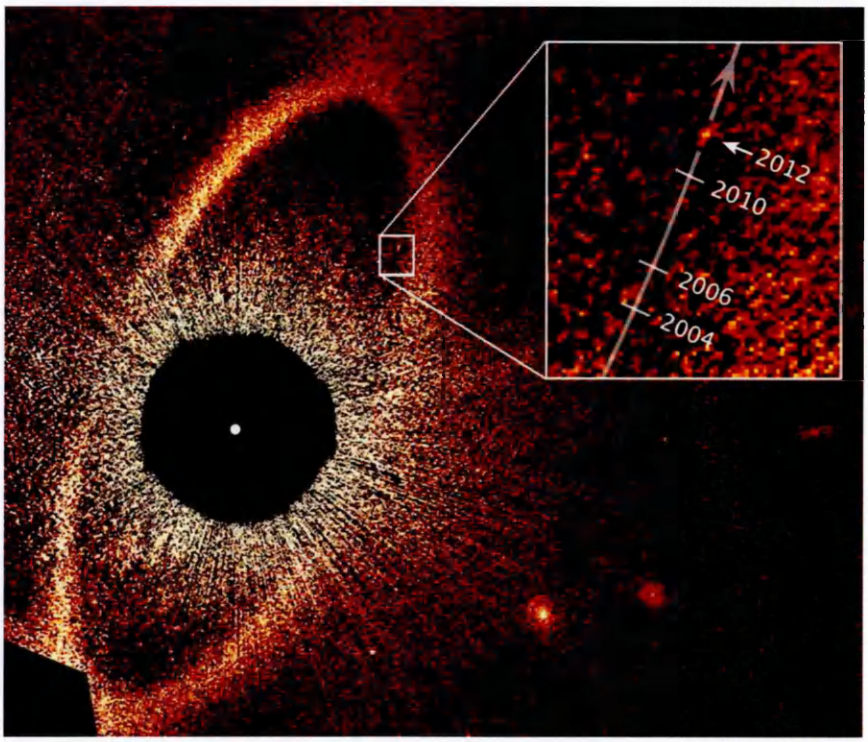


Figure 5: HST (Hubble Space Telescope) coronagraphic image of Fomalhaut at 0.6 μm. Image credit: [http://www.nasa.gov/images/content/717874main_p1301aw-orig_full.jpg]

This method was successful in the detection of Fomalhaut b. Figure 5 shows a false colour HST (Hubble Space Telescope) image of the Fomalhaut system where Fomalhaut b is made visible by light from its host star being scattered by an optically thick circum-planetary dust cloud (Kalas et al. 2013). Fomalhaut b is ~10^9 times fainter than Fomalhaut, orbiting at a separation of ~100 AU. Its wide 320,000 day orbit has been tracked by imaging over the course of many years. Because of the circumstances required for this method its success is more limited compared to other methods, responsible for only 63 (exoplanets.eu, Dec 2015) of planets found so far.

1.2.4 Astrometry

Astrometry is the science of accurately measuring the positions of stars. When two astronomical bodies are gravitationally bound they will orbit about a common barycentre. This gravitational interaction between the two bodies will induce a wobble in their positions as seen from Earth. Using high precision astrometry it is possible to accurately measure the positions of stars and in doing so characterise this wobble and directly infer the masses of both bodies involved. Should a star have one or more orbital companions then this induced wobble will take a regular form over time.

In multiple planet systems, the many gravitational influences will have a complex effect on the relative position of the barycentre over time. These overlapping rhythmic perturbations can be identified in the complex wobble of the star over time. By differentiating these collaborative perturbations from one another, previously unseen planets can be detected. The resulting amplitude of astrometric displacement of a star's reflex orbit is given by the relationship in Equation [3].

$$A = 3\left(\frac{M_P}{1\,M_\oplus}\right)\left(\frac{a}{1\,AU}\right)\left(\frac{M_*}{1\,M_\odot}\right)^{-1}\left(\frac{D}{1\,pc}\right)^{-1}\mu as \qquad [3]$$

where A *– amplitude of astrometric displacement* a *– semi-major axis*
M_P *– planet mass* M_* *- star mass* D *– distance*

In Equation [3] A = amplitude of astrometric displacement, a = semi-major axis, M_P = planet mass, M_* = star mass and D = distance. From Equation [3] we can see that the astrometry method is most effective for finding high-mass planets in wide orbits around nearby relatively low-mass stars. The size of the astrometric wobble increases as the planet's semi-major axis increases, so too does the orbital period in accordance with Kepler's third law ($P \propto a^{3/2}$). The mass ratio between the star and the planet is important because high mass planets will exert a greater pull on their host star and even greater the

lower the mass of the star. Astrometry can thereby detect relatively small planets orbiting far from their stars, as is the case for Earth, providing a crucial advantage for scientists looking for Earth-like planets in habitable zones.

There are also many limitations to the astrometry detection method which have made it difficult to implement, resulting in no planetary discoveries to date. Astrometry by its very nature is highly sensitive to the distance of the stars being observed from the detector and requires a high degree of precision to detect the subtle stellar motions. The attainable precision for ground based telescopes is further limited by the effects of atmospheric distortions, a huge obstacle. Most ground-based telescopes are not capable of measurements less than ~0.5 as (arcsec), in comparison the space based HST has a resolution of ~0.1 as. Using equation [3] it can be determined that from an observing distance of 10 pc the amplitude of astrometric displacement of the Sun due to Jupiter would be 0.5 mas, and for Earth 0.3 μas. Such precisions are not possible with HST.

Launched on 19 December 2013, the Gaia spacecraft (ESA) with a precision of 10 μas is expected to detect tens of thousands of Jupiter-sized planets beyond the Solar System. Using Equation [3], this precision would be capable of detecting a Jupiter analogue at a distance of ~500 pc.

1.2.5 Timing variations

It is also possible to detect the presence of planetary bodies by the effects they have on timing events within their host system. The earliest discoveries of exoplanets of any kind where due to the time variation effects they had on their host pulsar, in the case of PSR 1257 12 b, c and d, discovered in 1992 (Wolszczan & Frail 1992). In this scenario the

usually regularly observed pulses are offset by the gravitational influence of the orbiting bodies.

Timing variation effects can also occur in the expected transit times of known planets caused by unseen companions. Although the dominant source of gravity in any stellar system will be due to the host star, all other massive bodies will be exerting their own gravitational influence leading to perturbations in one another's orbits to some extent. This can result in changes to the orbital periods and inclinations of other bodies giving rise to transit timing variations (TTVs). Through careful timing of known transit events, the existence of additional planets can be inferred from the timing effects they cause.

The amplitude of the TTV effect on one planet's orbital period will be proportional to the mass and separation of the object causing it. It is therefore possible to infer mass of the effecting object using Kepler's laws. The extent of these TTVs will be much greater if the bodies involved are in resonant orbits where the orbital periods of two planets have an integer ratio. The planets involved will be in conjunction at a regular frequency and position, thus amplifying their gravitational impact. Such orbits can have a greater effect on the orbits of each planet as well as the entire system. The Kepler planetary survey has uncovered many planets from their transits and ongoing work on the light curves of those previously confirmed is uncovering additional system planets via TTVs (Steffen et al. 2013).

"The transiting planet Kepler-19b shows transit-timing variation with an amplitude of 5 minutes and a period of about 300 days, indicating the presence of a second planet, Kepler-19c, which has a period that is a near-rational multiple of the period of the transiting planet" (Ballard. S et al. 2011)

This method could also be applied to the detection of exo-moons as the shared barycentre would be orbiting the host star imposing a regular TTV on the midpoint of transit.

1.2.6 Microlensing

"Matter tells space how to curve, and Spacetime tells matter how to move" –John Wheeler

All gravitational bodies cause a local curvature of space time which causes deviation in light passing by. The extent of the light curvature is a direct result of the mass of the planet which although is not observable by its own light, is detectable by the effect it has on the background light of distant stars. As a distant planet moves across a starry background the space time curvature it imparts acts as a lens and magnifies the distant background starlight. Additionally if a planet happens to align with the Einstein ring of its host star it will have the effect of enhancing the star's magnification effect for a short time. These momentary increases in light can be observed from Earth implying the planetary presence.

This effect requires a momentary chance alignment between the distant starlight and the lensing planet in the observer's line of sight. The chance of such alignments is

vanishingly small (~10^{-6}) for any individual star but will be greater in the more densely populated regions and hence the region best studied for this effect is the Galactic bulge. To further increase the probability of detecting these events modern technology allows continuous monitoring of millions of stars aiding in the detection of such events. However, this method has a serious disadvantage as lensing events are short-lived one-off occurrences, there is no opportunity for follow up observations making confirmation of planetary properties highly unlikely.

The majority of known exoplanets have been discovered using the RV or transit methods, both of which have a bias for planets that are relatively close to their host star. As we can see from looking at our own Solar System most of the planets are found in much further out orbits than transit detection or RV would allow. If this is typical we would expect there to be an up-tapped larger population of further out planets that gravitational lensing could probe.

"This method is even able to detect planets not bound to any host star and recently a population of planets of this sort was discovered by microlensing." - T. Sumi et al. (2011)

Additionally gravitational microlensing is the only method for which the detection of planets outside the Milky Way is conceivable.

1.3 State of the Galactic Population

As the rate of discovery continues to increase and from what we have learnt about the abundance of planets in our Galaxy it is clear that we are not close to exhausting the

potential for discovery. For this reason, and because our knowledge of key phases in the processes of planetary formation and evolution is incomplete, the field of exoplanet research is likely to remain observation-led for years to come.

As technology advances we are ever more able to discern subtle signals which were previously unattainable, and thanks to advanced observational techniques the sensitivity limitations are being pushed ever further. Although advanced techniques are available they are the reserve of space telescopes and the few advanced ground based telescopes able to rival their space based counterparts. By their nature these high capability telescopes are in high demand, limiting their availability in time and expense to most projects. This means that the majority of exoplanet investigations capable of in-depth characterisation are significantly constrained to those planets which are orbiting nearby and bright stars.

At this time the true population of planets in our Galaxy is unknown. Only select regions of the Galaxy have been surveyed, and the locations of the planets so far discovered have been influenced by the survey region and detection biases of particular survey projects. For example, Kepler is a pencil beam survey and hence has only made detections within a particular small field. In contrast SuperWASP is a wide field survey and takes in the majority of the night sky thereby detecting planets in all directions.

From the promising results of recent years planets appear to be very common with the number of planets now thought to outnumber the stars in our Galaxy. A recent estimate stating that, on average, every star in the Milky Way has 1.6 ($\pm^{0.72}_{0.89}$) planets (Cassan et al. 2012). Assuming the total number of stars in the Milky Way to be at least 10^{11}, that

would put the number of bound planets in our Galaxy at ~ 2 x 10^{11}, a drastically different figure to the 8 known for 150 years prior to 1995.

"We conclude that stars are orbited by planets as a rule, rather than the exception." – *(Cassan et al. 2012)*

Exoplanetary science has produced many completely unexpected results. There is an astonishing diversity of planet types and arrangements including those so exotic that they have no Solar System analogue. Planets have been found in wide orbits taking hundreds or even thousands of years: e.g. *Oph 11 b*, P = 1998 yr, *Fomalhaut b*, P = 876 yr (exoplanets.eu).

Planets have been found around binary stars: e.g. *Kepler-413(AB) b* (Kostov et al. 2014). Rogue planets have been detected in the cold reaches of interstellar space without a host star: e.g. *PSO J318.5*-22 (Liu et al. 2013). These unbound planets are thought to have been ejected from their parent system by substantial gravitational perturbations. Only a few cases of this type of planet have been found, but given their proposed origin, being ejected from a planetary system, they are expected to be common. Some estimates put the number being greater than the number of stars in the Galaxy (T. Sumi et al. 2011). Taking Sumi's estimate of 1.8 ($\pm^{1.7}_{0.8}$) times as many rogue planets as stars, this brings the estimated Galactic population to be between ~1 x 10^{11} and ~5 x 10^{11} planets.

Several planets have been found within the habitable zones of their host, this being the temperature region where liquid water can exist on their surfaces, a staple of life as we know it. So far our own planet provides the only example of life currently detected

anywhere in the Universe but given the vast numbers of exoplanets, it is inconceivable that the conditions that are so amenable to life here on Earth will not be replicated elsewhere in the Galaxy.

Hot Jupiters

Some giant planets have been found in much tighter orbits than anything in our Solar System, being heavily irradiated by their host star. These so called "Hot Jupiters" are gas giants similar to our own Jupiter (0.3-13 M_{Jup}) found in tight orbits, between approximately 0.015 and 0.5 AU. Consequently they are much hotter than they should be from their own internal processes.

"Another factor contributing to the extreme heating of the irradiated face of the planet is that HJs are expected to be tidally locked, due to the relatively short time-scale for tidal synchronisation compared to the age of these systems." - Knutson et al. (2010)

The numbers of hot Jupiters being discovered was a huge surprise to astronomers. These findings challenged the traditional expectations of where a planet like this could be found, as it is extremely unlikely that a planet could form and persist in such an extreme environment. It is suggested that hot Jupiter planets must have migrated inwards by some means. The high levels of insolation cause these planets to become more 'bloated' and less dense then they would otherwise be.

Due to their extreme circumstances hot Jupiters are not expected to be typical for the Galactic population and are only estimated to represent ~1.2 (± 0.38) % of planets (Wright et al. 2012) in the Milky Way. However with their large size and location hot

Jupiters are most favourable for most detection techniques and represent a significant fraction of the early exoplanet discoveries, approximately 20% as of 2012 (Wright et al. 2012). This proportion will likely change in the future with the advent of projects with improved sensitivities able to reveal the smaller mass planets believed to be more typical of the Galactic population.

2. Mass losing Planets and the Case of WASP-12

The issue with Composition

While much progress has been made in the last 20 years on the discovery of exoplanets, for the vast majority of exoplanets only their most basic properties such as their mass, size and orbital parameters are known. It remains very difficult to infer composition. Hypothesised compositions of planets have been based on assumed formation models, their densities, as determined by their mass and radius, as well as the spectroscopically observed metallicities of host stars.

Inferring planetary composition from host star metallicity is problematic as the temperatures within the star will be greater than the dissociation temperatures of certain chemical compounds making it impossible for compounds which are likely to compose the planet to exist. Additionally these close in mass losing planets experience higher UV fluxes than Solar System gas giants, likely to alter their photochemistry (Knutson et al. 2010). The inherent difficulty with inferring composition based on density estimates is that several very different structures and compositions can match a particular mass and size (Rogers L.A. & Seager S. 2010.), as shown in Figure 6.

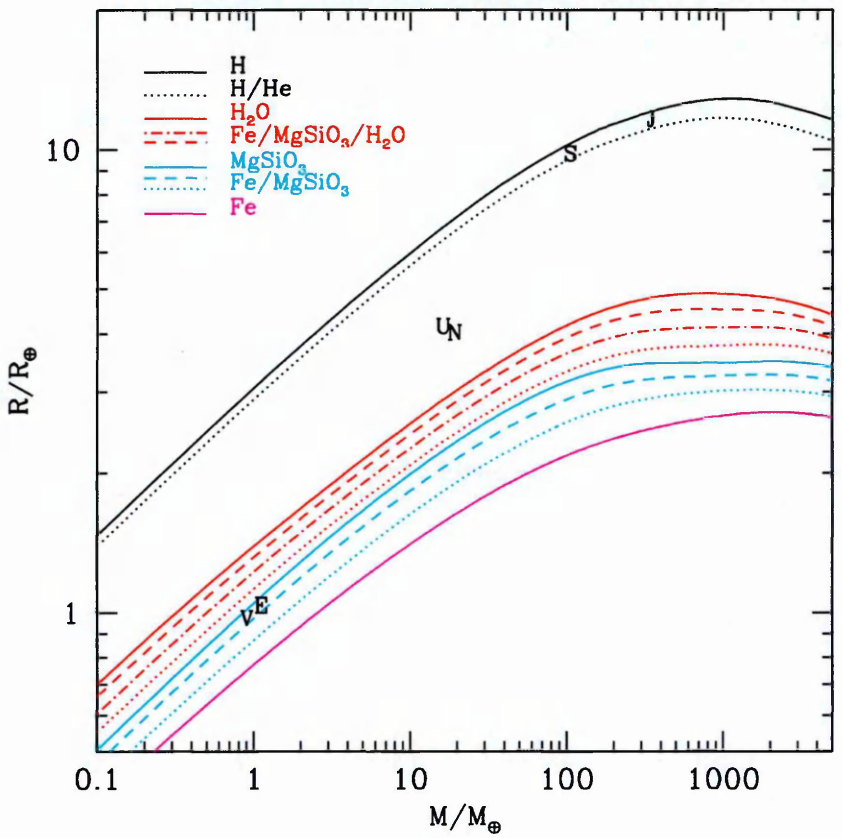


Figure 6: This plot shows the mass-radius relationships for a variety of different planetary compositions. The compositions of each coloured line are given by the legend in the top left corner. The solid and non-solid lines represent homogeneous and differentiated planets respectively (exact chemical ratios of the differentiated planets can be found in S. Seager et al. (2007)). The locations of the Solar System planets in this regime are denoted by their initials on the plot. Image credit: (S. Seager et al. 2007)

The problem of inferring planetary composition from density is especially true of rocky bodies, who in comparison to their host stars and gas giant counterparts are much more chemically complex comprising rocky material which is an aggregate of minerals and other compounds. In order to accurately reveal mass-radius-composition relationships of the various planet classes it is crucial to make direct measurements of bulk compositions. As it would not be easy to probe to any meaningful extent below their respective surfaces, another investigative approach is required.

Mass-Loss

As mentioned in Section 1.3, planets have been found in extremely close orbits around their host star. As a consequence of their proximity, some are expected to undergo mass-loss via irradiative ablation, with the rate of mass-loss being powered by the stellar extreme-ultraviolet flux (Lanza 2014). The high effective surface temperatures on these planets make their atmospheres 'bloated', with radii larger than would be expected in less extreme conditions. For instance the Sun like star WASP-68 hosts a close-in hot Jupiter planet, WASP-68b, which has a mass of 0.95 (± 0.03) M_{Jup} but a radius of 1.24 ($\pm\ ^{0.1}_{0.06}$) R_{Jup} (Delrez et al. 2014). Similarly another Sun like star WASP-48 hosts the hot Jupiter, WASP-48b, with a mass of 0.98 (± 0.09) M_{Jup} and a radius of 1.67 (± 0.10) R_{Jup} (Enoch et al. 2011). If these extended atmospheres reach beyond the planet's Roche lobe it will result in diffuse material pouring out into the orbital plane. Even in the cases of extreme proximity rocky planets the surface temperatures can be hot enough to sublimate material of the planet's surface, which is then thrown up into the atmosphere and lost to the orbital plane (Rappaport et al. 2012; Perez-Becker and Chiang 2013).

Opportunity for Transmission Spectroscopy

These diffuse clouds of material present a unique opportunity to go beyond atmospheric characterization and sample the bulk composition. As mentioned previously (Section 1.2.2), it is possible to perform spectroscopy on planetary atmospheres for transiting planets, and hence perform detailed chemical characterisation. Using the same technique it would be possible to examine these diffuse exospheres as the escaping material crosses the disc of the star in our line of sight. These catastrophic planets can therefore serve as good 'laboratories' to test and constraint models of exoplanet atmospheres and dynamical evolution (Fossati et al. 2011).

The WASP-12 System

One of the most extreme cases of this phenomenon observed to date, is that of the WASP-12 system, parameters of which are shown in Table 1.

Parameter	*WASP-12*	*WASP-12b*
Distance	*427.0 ± 90.0 pc*	-
Age	*1.7 ± 0.8 Gyr*	-
Spectral type	*G0*	-
Magnitude, m_v	*11.69*	-
Effective Temperature, T_{eff}	*6300.0 ± 150.0 K*	-
Mass	*1.35 ± 0.14 $M_{\odot}$*	*1.404 ± 0.099 M_{Jup}*
Radius	*1.599 ± 0.071 $R_{\odot}$*	*1.736 ± 0.092 R_{Jup}*
Semi-major axis	-	*0.02293 ± 0.00078 AU*
Orbital Period	-	*1.0914 day*
Eccentricity	-	*0.0 ± 0.01*
Inclination	-	*86.0 ± 3.0 deg*

Table 1: properties of the WASP-12 system. Source: [exoplanets.eu]

At an orbital distance of just 0.023 AU, corresponding to just ~2 stellar radii from the surface of the star, WASP-12b has a period of just 1.09 days. Figure 7 shows this arrangement to scale. This proximity results in an effective surface temperature of 2516 K (Hebb et al. 2009). This temperature is far greater than the melting point of iron, 1811 K, and all silicates, <1475 K, making WASP-12b one of the most irradiated and 'bloated' planets known. It was therefore expected to be experiencing ongoing mass-loss. Observations of Fossati et al. (2010b) and Haswell et al. (2012) confirmed that this was the case, with Haswell et al. producing an early estimate for the mass-loss rate of 10^{-7} M_{Jup} yr $^{-1}$.

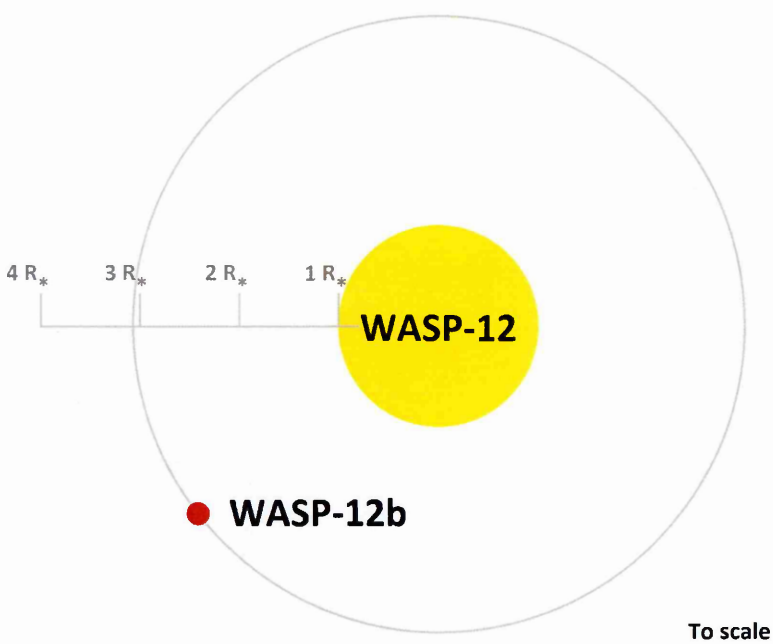


Figure 7: WASP-12 system shown to scale. WASP-12: radius 1.59 $R_{\odot}$, WASP-12b: radius 0.174 $R_{\odot}$, semi-major axis 4.93 $R_{\odot}$.

Absorption anomaly

Further investigation of the spectra of the host star WASP-12 revealed the surprising result of a complete lack of emission in the cores of the Mg II h & k and Ca II H & K lines throughout all phases of the planetary orbit (Knutson et al. 2010; Fossati et al. 2013b; Haswell et al. 2012). This result was surprising since Ca II and Mg II usually present very strong emission cores in the stellar spectra of active main sequence stars of the age and spectral type of WASP-12.

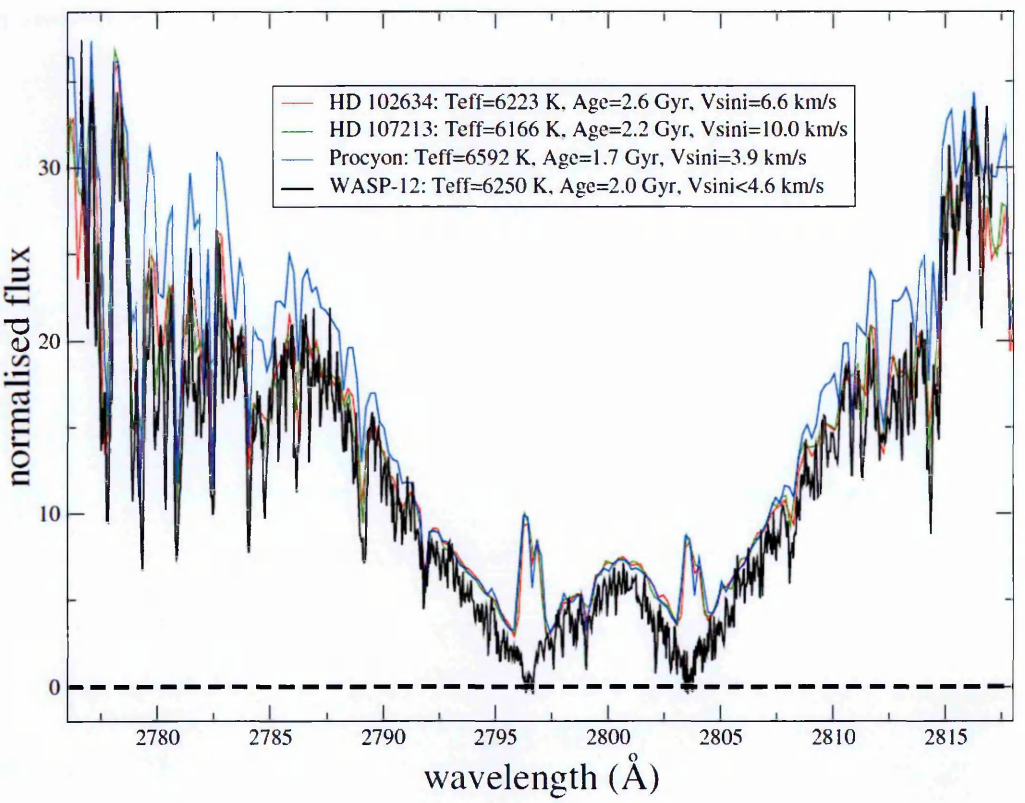


Figure 8: comparative spectra for WASP-12 and three comparison stars in the region covering the Mg II resonance lines. The stars HD 102634 and HD 107213 are of similar age and effective temperature to WASP-12. The star Procyon is known to have low activity level. Image credit: Figure 16 (Haswell et al. 2012)

Figure 8 shows the stellar spectra in the region of the Mg II h and k resonance lines for WASP-12 and three other stars of similar age and temperature. This near ultra-violet

wavelength region is dominated by the photospheric absorption of the Mg II doublet, giving rise to the distinctive broad 'W' shape. It can be seen that the line profiles of the stars HD 102634, HD 107213 and WASP-12 (red, green and black respectively) are in good agreement for the majority of this wavelength region, as expected for such similar stars. However, they diverge drastically close to the emission cores centred at 2802.7 Å and 2795.5 Å. In these regions the line profiles of the stars, HD 102634, HD107213, and Procyon, exhibit these anticipated distinct emission peaks. In contrast WASP-12 exhibits anomalously low flux with steep gradients decreasing towards the emission cores which themselves exhibit zero flux.

The Mg II h and k and Ca II H and K emission cores are normally prominent features in the stellar spectra of active stars. This is due to the abundance of these elements in most stars twinned with their low first activation energies making them strong emitters and absorbers of photons at these specific wavelengths. These elements also exist in interstellar gas and dust clouds. Here they will be in their ground states and will readily absorb photons, therefore not much material is needed for strong absorption features to occur. This makes these line cores sensitive probes for material absorption in stellar spectra, whether from the interstellar medium or from within a system.

The observation of total absorption in these line cores is consistent with vast quantities of these elements blocking the emission lines in this region. This suggests the surprising result of an extensive exosphere enshrouding the entire WASP-12 system supplied by the evaporating planet.

Absorbing Gas Around the WASP-12 Planetary System (Fossati et al. 2013b)

To confirm that this is indeed the case L. Fossati et al. (2013b) considered three scenarios which could be causing this effect.

1. WASP-12 was a stellar anomaly uniquely inactive in these wavelength regions.
2. The extrinsic absorption is due to the density column of the interstellar medium (ISM) in the WASP-12 line of sight.
3. That it is the case that the absorption features are being caused by enshrouding gas within the WASP-12 system.

Stellar activity

WASP-12 is a 1.7 Gyr old main sequence star of spectral type G0 so it is considered unusual that these spectral emission lines are not observed considering they are usually strong for similar stars of that age. The WASP-12 Ca II H and K line profiles were compared to those of two other stars with similar stellar properties and found to be significantly deeper. Previous work by Knutson et al. (2010) measured the chromospheric activity index, $\log R'_{HK}$ (see Section 4.2), for 50 stars hosting transiting planets including WASP-12. They found WASP-12 to be by far the least active with a $\log R'_{HK} = -5.500$, falling far below the 'basal' level of activity expected even for old inactive main sequence stars, $\log R'_{HK} = -5.1$ (Henry et al. 1996; Wright 2004). Given the spectral type and probable age, of WASP-12 stellar inactivity alone cannot account for the anomalously low activity in the Mg II and Ca II resonance line cores.

ISM

Next the ISM was considered. Haswell et al. (2012) used the expected Mg II k-line absorption due to ISM reported in the line of sight to the star α Cen A, to predict a plausible column density in the line of sight to WASP-12. This indirect estimate of the ISM column density was found to be a factor of 10 less than what would be required to reproduce the Mg II k-line absorption seen in WASP-12. Thus an interstellar origin for the absorption was considered unlikely. Following this, Fossati et al. (2013) obtained high resolution spectra of three bright fast rotating stars in the WASP-12 field so that the ISM column density in the WASP-12 line of sight could be directly measured. Their recorded value was found to be in agreement with that proposed by Haswell et al. (2012), and far below the value required to cause the absorption seen in the WASP-12 Mg II line cores.

Circumstellar dust

Thus it was established that the absorption of these spectral lines must be due to material within the WASP-12 system itself. This indicates that for the observation of zero flux in these regions throughout all phases of planetary orbit that the diffuse material must form a stable and thick circumstellar cloud enshrouding the system and continuously 'veiling' the stellar disk, being fed by gas escaping from the heavily irradiated planet.

Similar Cases

Similar anomalous absorption effects have been seen for stars X0-4, CoRoT-1, WASP-13, WASP-17 and WASP-18 (Fossati et al. 2013b) all of which host close-in planets. These systems are believed to be undergoing the same phenomenon as WASP-12. These

hot Jupiter hosting stars are highlighted by green circles on Figure 9. WASP-12 is represented by the blue triangle.

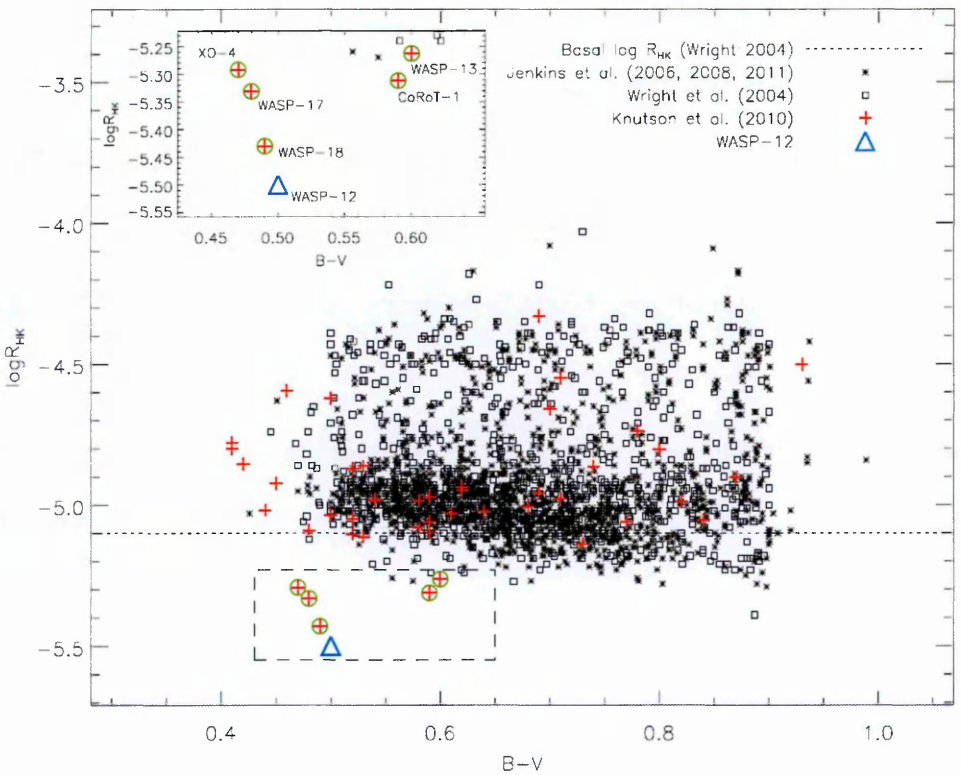


Figure 9: Plot shows stars in a B-V vs. log R'_{HK} plane. The dotted line indicates the main sequence basal limit log R'_{HK} = -5.1. WASP-12 shown as the blue triangle falls well below the basal flux line with log R'_{HK} = -5.50. Other known planet hosts with anomalously low log (R'_{HK}) are highlighted by green circles. The outlier region emphasised by the dashed box is shown enlarged in the top-left corner. Image credit: Figure 2 (Fossati et al. 2013).

Figure 9 shows log R'_{HK} vs spectral colour (B-V) for a population of stars from 5 activity catalogues. The activity index log R'_{HK} parameterises the flux in the Ca II H & K line cores vital to the investigation of this mass-loss effect. The dotted line indicates basal level of activity, log R'_{HK} = -5.1, mentioned previously. The only stars expected to fall below this line must be either evolved sub-giants at the end of their main sequence phase, or stars which are subject to the veiling effects of circumstellar gas, as is the case

for WASP-12. A subgroup of outlier low activity stars is emphasised in the dashed box in the bottom left of Figure 9. This region is also shown magnified in the top-left corner of the plot. The five stars highlighted with green circles are known hot Jupiter hosts whose low $\log R'_{HK}$ values may imply similar circumstellar gas shrouds to that of WASP-12. Therefore the observation of anomalously low $\log R'_{HK}$ may indirectly infer the presence of close-in evaporating planets.

Of those hot Jupiters hosts highlighted in Figure 9, WASP-18 bears the most similarities to the WASP-12 system. Indeed further investigation by Fossati et al. (2014) concluded that the WASP-18 system is very likely surrounded by a circumstellar cloud, made of material lost by the planet. The confirmation of another system hosting a close-in planet with anomalously low $\log R'_{HK}$ strengthens the hypothesis of circumstellar absorption due to the presence of mass-losing planet, as a cause. Other than increasing the sample size an advantage of studying the WASP-18 system is that it is 2.3 mag brighter than WASP-12, allowing for more precise measurements of spectra and system properties. Hence WASP-18 is a key target for future studies of evaporating hot Jupiters.

As well as the large hot Jupiter planets, close-in disintegrating small rocky planets have also been observed. Lost material from these planets would also likely comprise Ca II and Mg II and thus impose the same absorption effects on the stellar spectra, although to a much lesser extent. KIC 12557548b is a small evaporating rocky planet losing material from its surface forming a comet-like tail (Rappaport et al. 2012). It is thought that KIC 12557548b and similar small rocky bodies could be the remnant cores of extreme hot Jupiters, representing the final stages of the process studied in WASP-12b. If this is the case it presents the unique opportunity to sample the composition of a giant planet core

usually hidden away beneath a deep atmosphere. Otherwise for the case of a small rocky planet having migrated catastrophically close to their host star, the opportunity to measure the composition of this planet type is presented.

Model Development

The relationship between close-in evaporating planets and anomalously low $\log R'_{HK}$ presents the possibility of indirectly detecting exoplanets from existing archival spectra. If existing archival spectra were to be parameterised in terms of $\log R'_{HK}$ then promising outliers falling below the basal activity limit could be identified as potential mass-losing planet hosts. With candidates identified subsequent investigation could confirm if they are hosting a mass-losing planet. This would have the potential to identify gas shrouds in known planetary systems or even to uncover previously undetected mass-losing planets which may have eluded detection by traditional methods. If a planet was first discovered in this way this would prove to be a whole new method of detection. There may well be a bounty of planets yet undiscovered hidden within existing stellar archives, even those not intended for planet detection. Finding a greater number of mass-losing systems could offer new insights on the processes expected to be contributing to mass loss and also look for correlations that allow us to identify the processes driving planetary formation and evolution. These disintegrating exoplanets are ideally suited for follow-up work, focused on chemical composition, not possible except for under these extreme circumstances.

Identification and study of a significant number will allow for the testing and development of planetary evolution models. This will constrain how many of these close-in remnant planets we are likely to find which can then be used to inform future follow

up work. In addition a larger sample size will allow proposed models to be tested, such as the relationship between surface gravity (log g) and chromospheric emission (log R'_{HK}) (Lanza 2014).

These catastrophic end of life planets are expected to be relatively rare, however as discussed in (1.1.3), hot Jupiters represent ~ 1.2 (±0.38) % (Wright et al. 2012) of the planets in the Milky Way which still gives a number of ~2 x 10^9. It is therefore not unreasonable to expect these extreme cases in the spectral archives.

The possibility of planets exhibiting these features hidden in existing archives and eluding traditional detection methods proposes the fantastic opportunity for a host of new exoplanet discoveries. It is therefore important to develop a means of efficiently combing through this archival data and identifying these features of interest.

3. Spectroscopy

3.1 Definition and Application

Essentially all information that we can get from distant celestial objects comes in the form of light as it reaches us, so it is no wonder that the study of light (spectroscopy) is such a standard tool in astronomy. Using traditional telescopes we can examine the light from distant astronomical bodies allowing us to determine their exact positions and sometimes shapes. When this composite light is dispersed to form a spectrum of its component wavelengths, a whole host of information about the properties of the source object can be obtained by careful spectroscopic analysis.

From quantum mechanics we know there is a direct relationship between atomic transitions and electromagnetic radiation. All chemical species, be they atoms, ions or molecules have a discrete series of quantised energy levels. Species can transition between these energy levels either by absorbing a photon and jumping to a higher energy level, or emitting a photon and dropping to a lower energy level. In either case the photon associated with the transition will have the exact energy corresponding to the difference between energy levels. The wavelength associated with the energy of a photon is defined by the Planck relationship ($E = h\nu$).

The wavelengths of all transitions appear as distinct spectral lines within the spectrum of the object being observed. The series of transitions characteristic of each species forms a unique 'fingerprint' of spectral lines. A continuous spectrum, in which there are many energy levels with extremely small separations between them, emerges from deep within the hot interior of all stars. This continuum approximates a black body curve with the temperature of the star. As these continuum photons emerge through the cooler outer layers of the star, the photon energies characteristic to the atoms present in those layers

are absorbed. The spectrum of the star thereby appears as a continuous black body curve with dark spectral lines imprinted on it due to photospheric absorption at specific energies. Emission lines can also be seen in stellar spectra indicating energetic processes occurring within the stars outer layers. The composition of stars can be deduced by identifying the unique spectral barcodes of chemical species inferring their presence. The abundance of each species can be determined from the relative intensity of the transitions being observed. Observed differences in these spectral features can be used to give additional information of stellar motion (via Doppler shift), pressure, and even the effects of magnetic fields at the source.

3.2 ISM Absorption

If material such as cosmic gas and dust exists in the line of sight to a distant star, it will have the effect of imprinting additional absorption features on to the stellar spectrum. Providing that the medium is sufficiently optically thin that the character of the star can be identified, the extent of the increase in absorption at specific wavelengths will be proportional to the column density (number of absorbers). The presence of excess absorption of specific spectral lines can reveal the composition of the intervening medium.

3.3 Échelle Spectroscopy

In traditional spectrographs a dispersing element, typically a diffraction grating or a prism, is used to produce the spectrum. This produces a single spectrum which can be imaged using a suitable image sensor chip, typically a charge-coupled device (CCD). This spectral data can then be processed and analysed using computational techniques.

In this arrangement the wavelength range being captured is limited by the size of the CCD. As the spectrum will fall as a long thin strip across the CCD, the spectrum itself will only use a small amount of the full detecting area thus limiting the possible wavelength coverage of the instrument. One method for utilising more of the available detector area is to use an Échelle spectrograph. In an Échelle spectrograph this spectrum is further dispersed using a special diffraction grating, called an Échelle, which breaks this continuum into a series of stacked spaced orders like rungs on a ladder ('Échelle' being the French word for ladder), see Figure 10. This results in a spectrum of very high dispersion, but covering only a small wavelength range per order.

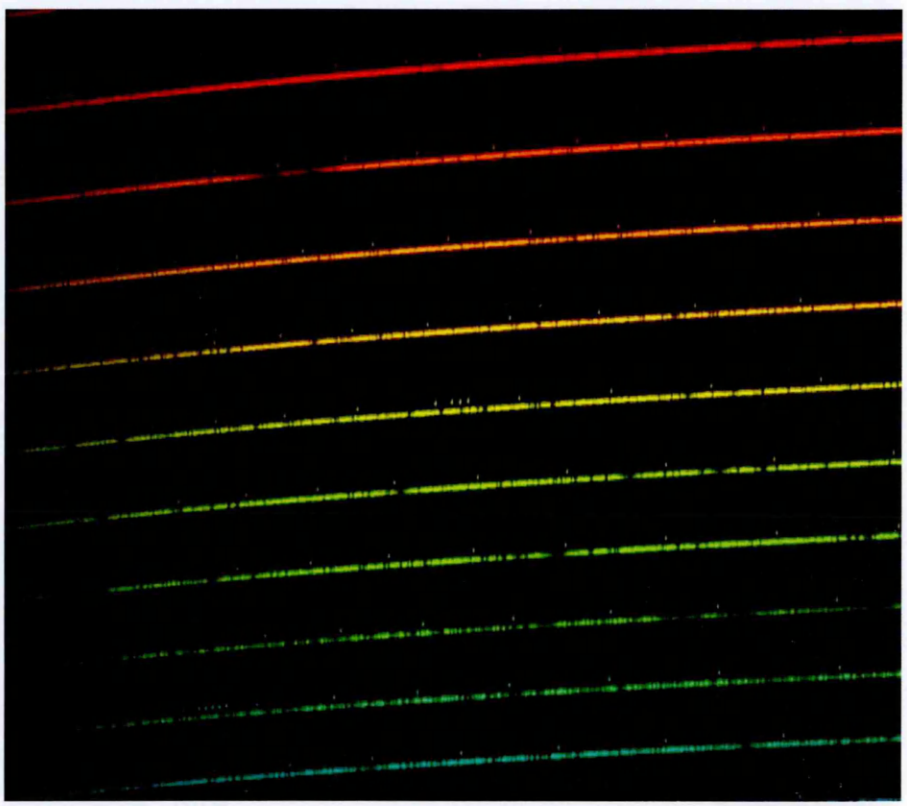


Figure 10: small section of an Échelle spectrum shown in colour. Stellar spectrum arranged as stacked orders. Simultaneous calibration spectral orders can be seen just above each stellar order. Image Source: http://www.giga-parsec.de/

These instruments offer both a reasonably high resolution and wide wavelength coverage, but at the expense of a comparatively more complex data reduction procedure. The inherent complexity of data reduction is a consequence of how the continuous spectrum has been broken up and stacked into separate orders on the same CCD chip. Complexity arises because there is more data to extract, the orders have a more complex shape, flat fielding is more difficult and dispersion of orders means more potential for cosmic ray interference. Also in some cases spatial overlap can occur between orders making background subtraction difficult. Fortunately, there are several dedicated software packages available designed specifically for handling these inherent obstacles in Échelle spectra.

The geometric distortion inherent to Échelle spectra adds additional complexity to the task of wavelength determination. To account for this, light from a calibration lamp source with a precisely known emission spectrum is simultaneously fed through the spectrograph. The calibration light Échelle orders are then projected onto the detector chip alongside the stellar spectral orders, as can be seen above each stellar order in Figure 10. The calibration light can be used to characterise the wavelength dependence on the geometric distortion of the spectral orders introduced by the spectrograph. In this way the Échelle spectra can be calibrated for wavelength.

Échelle spectrographs are most useful for chemical characterisation of stellar spectra and are used by telescopes such as HARPS (High Accuracy Radial Velocity Planet Searcher, La Silla, Chile), the 1.93 m telescope at OHP (Observatoire de Haute-Provence), and numerous other astronomical instruments. The reasonably high resolution and wide wavelength coverage makes it easier to investigate the spectral signatures present in the

starlight continuum, thus making Échelle spectroscopy a useful tool for characterising absorption features in the spectra of mass loss systems.

4. Mg II and Ca II Resonance Lines and log R'_{HK}

There are a number of ions of astronomical importance that present strong spectral lines which can be used to study different stellar processes such as stellar activity, magnetic activity and temperature. In particular, the near ultra-violet (NUV) spectral region contains a rich set of resonance lines of numerous metals, including the strong Mg II h & k resonance lines.

Magnesium II h & k resonance lines	
h line	2802.7 Å
k line	2795.5 Å

Mg II h and k resonance lines are established measures of chromospheric activity (Haswell et al. 2012) and are usually bright in active stars. The NUV region is mostly blocked by the atmosphere therefore space based observations are required, making it difficult to obtain Mg II lines for many stars. Fortunately the Ca II H & K resonance lines present similar behaviour and lying in the optical region are much more amenable to study.

Calcium II H & K resonance lines	
H line	3968.47 Å
K line	3933.66 Å

The ground states of these ions consist of two optically active electrons outside of closed shells resulting in low first activation energies. Magnesium and Calcium have first ionisation potentials of 7.6 eV and 6.1 eV respectively, both significantly lower than that of Hydrogen (13.6 eV). These ions' low activation energies mean that they exhibit strong absorption lines in stellar spectra, and strong emission lines in the case of active stars. These emission lines are sensitive to the temperature, density, and velocity gradients of

the lower chromosphere and are often used as an indicator of stellar activity (Ayres 1979). They have therefore been studied by numerous projects. One such example is the long term Mount Wilson observatory HK project (Wilson 1978), which investigates stellar magnetic activity, ages and rotations of many stars using Ca II H & K lines. The Ca H and K lines present as deep broad absorption lines in stellar spectra, with centralised core emission reversals for chromospherically active stars. Figure 11 shows two spectra for both an active and inactive star, in the region of the Ca II H and K lines, where the emission peaks can be clearly seen.

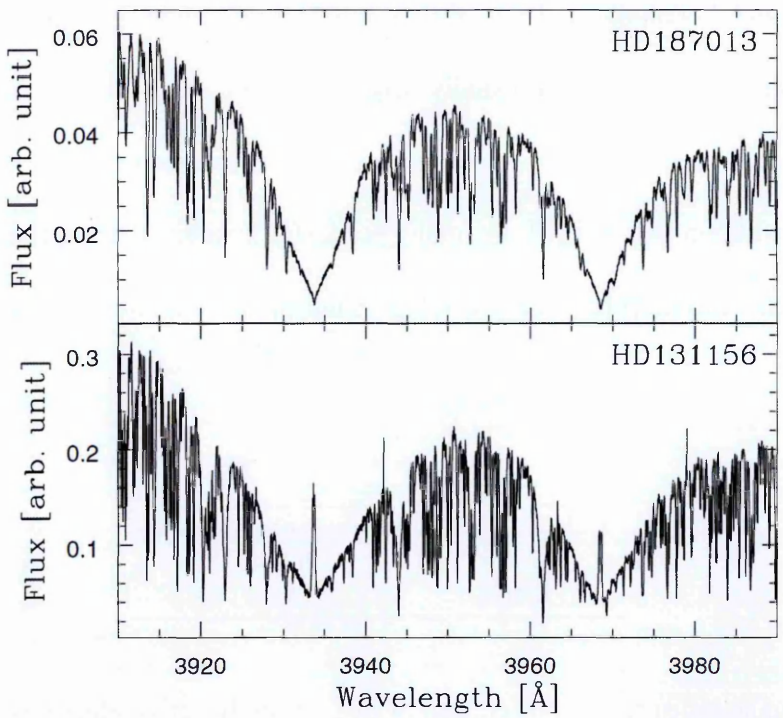


Figure 11: Two SOPHIE spectra in the region of the Ca II H and K lines, for the chromospherically inactive star HD187013 (top) and the chromospherically active star HD131156 (bottom).Image source: (Boisse et al. 2010)

4.1 Mg II and Ca II ISM absorption

As well as being strong absorbers and emitters within active stars the Mg II and Ca II

ions also act as strong absorbers within interstellar medium. Their strong absorption properties are also a result of their low activation energies. When found in diffuse gas clouds they exist in the ground state and will readily absorb photons, so less gas is needed to cause subtle but noticeable absorption in these same emission wavelengths. The expected abundance of Mg II and Ca II make their characteristic absorption features sensitive probes for the presence of absorbing gas. It is for this reason that these resonance lines are such sensitive probes for detecting shrouding gas and dust in mass losing systems (Haswell et al. 2012).

4.2 Ca II H and K activity, S-index

The established method of recording the Ca II H and K activity is using the Mt. Wilson survey's S-index, as described by Vaughan et al. (1978). The HKP-2 photometer mounted on the Mt. Wilson 60-inch telescope (see Duncan et al. (1991) for details of the observational setup), measures the fluxes in four bandpass channels H, K, V and R. The channels H and K measure the fluxes in two triangular bandpasses, with FWHM = 1.09 Å, centred on the H and K emission cores respectively. The channels V and R measure the fluxes at two 20 Å wide continuum bands located at 3991.07-4011.07 Å and 3891.07-3991.07 Å respectively. The V and R windows estimate the continuum on both sides of the Ca II doublet, sufficiently spaced to avoid the broad wings of the absorption. The S-value is then defined as the ratio between the chromospheric flux (H + K) and the Bolometric flux (V + R), as shown by Equation 4:

$$S = \alpha \frac{H+K}{V+R} \qquad [4]$$

In this equation α is a calibration constant with a value either fixed at 2.40 or 2.30 (Duncan et al. 1991), or regularly adjusted following calibration (Baliunas et al. 1995).

S-value is an index of Ca II H and K line flux unique to, and defined by, the HKP-2 instrumentation. As such for S-values to be measured using other telescopes, calibration to the Mt. Wilson scale is required. To achieve this, a set of 'standard' stars which are known to be actively stable must be observed with both instruments to establish a linear correction factor, S_{MW} (see Equation [5]).

S-index revised to R'_{HK}

Since the H and K fluxes used to compute the S-value contain both photospheric and chromospheric components, the S-value has an inherent colour (B-V) dependence. This means that, although the S-index may be useful for evaluating the activity of individual stars, comparing the activity of different stars, especially those of different spectral type, is a problem. This problem was overcome with the introduction of the dimensionless R'_{HK} index, by Noyes et al. (1984), which subtracts the photospheric component and normalises the chromospheric to bolometric flux ratio, thereby accounting for the colour dependence. R'_{HK} is defined by:

$$R_{HK} = C_{cf}(B-V) * S_{MW} \quad [5]$$

$$R'_{HK} = R_{HK} - R_{phot}(B-V) \quad [6]$$

where the conversion factor C_{cf} (B - V) and the photospheric contribution, R_{phot} (B-V), are related to the colour (B-V) of a star. The calculation of these values is described in Noyes et al. (1984). The ' in R'_{HK} denotes that the photospheric contribution has been subtracted. Therefore by converting from the S-index to R'_{HK} accounts for the colour dependence, and allows for direct comparison of the chromospheric activity between

stars of different spectral types. The most widespread use of this value is in logarithmic form, log R'_{HK}, to account for large differences.

5. SOPHIE

5.1 Location and History

As mentioned previously, a multitude of spectral archival data exists from numerous spectroscopic instruments. One such instrument is SOPHIE (Spectrographe pour l'Observation des Phénomènes des Intérieurs stellaires et des Exoplanètes), a high-resolution cross-dispersed Échelle spectrograph located at l'Observatoire de Haute Provence, France. It is housed in the second floor of the 1.93 m telescope building and connected to the in house telescope by two sets of 100 μm diameter optical fibre sets.

It is the successor to the ELODIE Échelle spectrograph, well known for detecting the first exoplanet around a sun-like star, 51 Peg. The instrument was decommissioned in and replaced by the improved SOPHIE in late 2006. SOPHIE has since continued and extended the comprehensive exoplanet characterising survey of its predecessor (Bouchy et al. 2009b) and has led to many discoveries in the field of exoplanet research. This instrument's main focus is on RV detections. It is often used for high cadence RV follow up of transiting planet candidates from various projects such as Super WASP (e.g. Cameron et al. 2007), HAQT (e.g. Bakos et al. 2007), and CoRoT (e.g. Bouchy et al. 2008).

5.2 Instrument

The SOPHIE spectrograph yields a spectra with 41 spectral orders, of which 39 are currently extracted, to obtain wavelengths between 3872 Å and 6943 Å. From a list of

orders on the OHP site (http://www.obs-hp.fr/) shows the wavelength range of each of the 39 usable orders, below are shown the orders containing the Ca II H and K resonant lines centred at 3968.49 Å and 3933.68 Å respectively.

ord	ech	llmin	llcent	llmax	
0	88	3872.4	3913.1	3950.2	
1	87	3916.7	3957.9	3995.5	Ca H Ca K
2	86	3962.1	4003.8	4041.8	

As mention above in Section 3.3 there is some overlap in wavelength coverage between adjacent orders. For the Échelle spectrograph produced by the SOPHIE instrument the Ca II H line core is found on both order 1 and 2. This needs to be taken into account in post analysis.

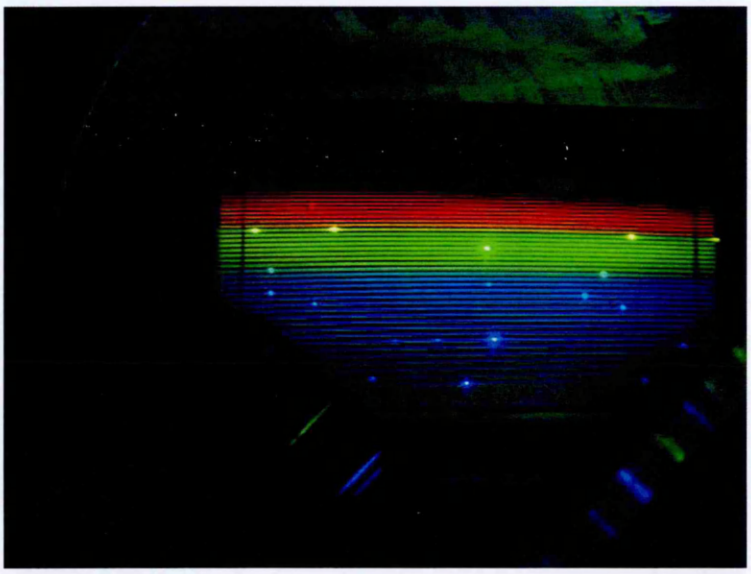


Figure 12: An Échelle spectrum as projected onto the CCD within the SOPHIE instrument. Image Source: http://www.obs-hp.fr/

A colour image of the resulting Échelle spectrum produced by SOPHIE can be seen in Figure 12. Here the spectrum is projected onto a translucent screen that approximates the area sampled by the CCD (4096x2048 pixels).

The setup includes two calibration lamps which are used to produce calibration frames needed for the consequent data reduction. A tungsten lamp is used for relative flux calibration ("flat field") and a thorium-argon lamp for wavelength calibration. Two fibre-pairs feed into the spectrograph from the telescope, each yielding different spectral resolution modes. One fibre pair is used for the high-efficiency, HE mode (R = 40000, where $R = \lambda/\Delta\lambda$), when higher throughput is desired in the case of faint objects. The other fibre pair is used to achieve the high-resolution, HR mode (R = 75000) which can yield accurate velocities down to 2-3 ms^{-1} (using simultaneous Th calibration), depending on the signal-to-noise ratio. Both fibre-sets consist of a fibre A and fibre B which can feed light from different sources simultaneously. Fibre A is normally used for the target while fibre B can be used to carry the sky spectrum, the simultaneous thorium-argon calibration lamp exposure or even be masked for no light exposure. An example of simultaneous calibration spectrum exposure can be seen in Figure 10.

5.3 The SOPHIE Data Archive

The SOPHIE spectrograph is entirely computer-controlled and features an entirely automatic data-reduction pipeline, which processes the data upon CCD readout. A summary of the SOPHIE operation process is given in the OHP report in appendices (Appendix 1). This automatic data reduction is performed by a dedicated workstation and software, adapted from the HARPS software, which includes bias subtraction, localization of the orders, optimal order extraction, cosmic-ray rejection, flat-fielding

and wavelength calibration. Information of the identification, calibration and reduction is written into the FITS file headers. All of the data from the automated data reduction pipeline, as well as the raw data from which it was produced, is stored in the SOPHIE archive (http://atlas.obs-hp.fr/sophie/) which is freely accessible to the online community. This extensive archive contains 73973 high quality spectra from 5634 objects (as of 19.04.15), 49855 of which are fully public, thus offering great potential for detailed investigation of their spectral features.

6. Recalibrating the SOPHIE Archive: Project Rationale

In this project, we are interested in investigating the phenomenon of close-in planetary mass loss. As stated before, this process has been observed and studied for just a few star systems so far. In order to better understand this phenomenon we would require a greater number of identified cases. We expect this phenomenon to be widespread, though not necessarily common, throughout our Galaxy and therefore we expect there to be a larger number of cases than are known so far. It is for this reason and the fact that stellar spectra have been obtained for a great number of stars that we believe many more examples could be harboured in existing archives.

In Section 2 it was shown how anomalously low measurements of $\log R'_{HK}$ could be an indicator of circum-stellar gas shrouds being fed by close-in evaporating planets. WASP-12 was the first star discovered to be hosting this phenomenon (Fossati et al. 2013b). It is shown in Figure 9 having low $\log R'_{HK}$ = -5.50 (Knutson et al. 2010) falling well below the lower limit expected for main sequence stars of $\log R'_{HK}$ = -5.1 (Wright et al. 2004). Likewise we expect similar systems to present as extreme outliers in a colour (B-V) vs. activity ($\log R'_{HK}$) plane, as WASP-12 does, demonstrated in Figure 9. A valuable undertaking for this line of investigation would therefore be to develop a procedure for accurately producing the $\log R'_{HK}$ for large numbers of stars with recorded archival spectra.

As described in Section 2, calibrating archival stellar spectra in terms of $\log R'_{HK}$ would allow for the simple comparison of stellar activity for many main sequence stars, in similar fashion to the work of Knutson et al. (2010). From the subsequent $\log R'_{HK}$ catalogues, anomalously low activity stars could be easily identified, as in Figure 9, on an archival scale. This would thereby aid the detection and investigation of mass-losing

planet systems. Additionally this method might prove a novel method of detecting previously undiscovered planets in existing archives.

As discussed previously (Section 2) the scenario of an evaporating close in planet results in specific absorption features in the stellar spectra, most notably in the cores of the Mg II and Ca II resonance lines. Whereas the Mg II lines lie in the NUV region, only covered by a limited number of telescopes, the Ca II lines are found in the optical region which is much better covered. The SOPHIE Échelle spectrometer, with a band pass of 3872 Å – 6943 Å covering the optical region, is therefore ideal for studying the Ca II H and K lines.

As mentioned (Section 5.3) the SOPHIE archive contains a plethora of stellar spectra freely available to the online community. Additionally being Échelle spectra they are amenable to investigation of specific spectral features. It is for these reasons and because we suspected that an important step (background subtraction) for our purposes had been skipped, that the SOPHIE archive will be the primary focus of this project. The high number of available archival stellar spectra gives us a reasonable chance of detecting these absorption features should they be common.

It is important to know when studying the Ca II H&K lines in SOPHIE spectra that they are located in a spectral region that suffers from background light contamination arising from internal scattered light and thermal sources within the instrument. Due to the way in which $\log R'_{HK}$ is computed, as detailed in Section 4.2, it is highly sensitive to the level of flux in the Ca II H and K line cores. It is therefore extremely important that any unwanted background light is effectively removed for the calculated $\log R'_{HK}$ to be

accurate.

The SOPHIE survey is not so concerned with low level flux, being primarily focussed on RV measurements. RV measurements require just the accurate measurement of wavelength shifts and hence it is not an issue for the SOPHIE archive that the recorded counts are artificially shifted up by background light contamination. The archive provides its own online reductions but is unlikely to perform the level of background subtraction sufficient for our investigations. As accurately characterising these low flux regions is so important to our investigations it was deemed necessary that we perform our own data reduction on raw archival data.

This project will focus on constructing a software pipeline to take unprocessed Échelle spectra and produce fully reduced Échelle spectrograms from which the orders of interest can be extracted for further study. This pipeline will make use of existing packages to perform each stage of the data reduction. However since effective background light subtraction is so integral to this work, new software was developed to perform this task. Developing a reliable method for background subtraction is essential step for generating reliable log R'_{HK} measurements that will be required in the future of these investigations.

7. Introduction to Data Reduction

The goal of this project was to construct a means of producing fully reduced spectral plots in the region of the Ca II H and K resonance lines from unprocessed Échelle spectra from the SOPHIE archive. Included in this process is the necessity to perform an effective background subtraction of the light contamination in this region. To get the data from the archive into a useable form it must first be put through a process of data reduction. This was done by performing a series of steps arranging the data and removing unwanted noise and systematic contributions to reveal just the underlying science data useful for our investigations.

As stated in Section 6 new software was developed to perform this critical background light subtraction. The effectiveness of the background subtraction will be evaluated by performing the full data reduction pipeline on a select number of target spectra, once using pre-existing background subtraction methods and then again using the new software.

8. Starfinder.py

The first step in the data reduction process was to consider which objects were recorded in the SOPHIE archive. This extensive archive contains the spectra of many astronomical objects including stars as well as Solar System planets and asteroids. A list of all of the objects contained within the SOPHIE archive was obtained by generating an online table using the archive sites search feature. From this table just the identifier column was extracted to an asci (.txt) file. Included in this list were all the stellar and non-stellar objects duplicated for how many times each target had been observed. In total this list, and hence the total archive (December 2014), contained 43,624 spectra (including those not yet made public). For this project we are interested in the stellar spectra and so it is important to categorise this list to reveal which of these objects are stars. To this end the script Starfinder.py was designed written in Python, the code for which can be found in full in the appendices (Appendix 2).

Python was selected as the language of choice, as it is a powerful and intuitive, object orientated programing language. Being freeware removes the obstacle of software licensing, so work could be easily imported and worked on using any machine, and not just those with the expensive and prohibitive license constraints. Also due to its popularity there is a wealth of online resources, and a significant online community of users. This made this language a sensible choice, as tools and resources for solving the specific problems would be readily available.

The script Starfinder.py worked on the SOPHIE archive spectra list file in three stages. First the function ***starfinder()*** removes any object names which are repeated within the file, leaving a list of unique object names. This revealed that there were 5795 unique objects in the archive. Next the function ***remover()*** removes any non-stellar object names

by referencing the text file 'remove_list.txt'. This file contains a list of known non-stellar objects whose spectra exist in the SOPHIE archive, including solar system planets and asteroids. Thus leaving a list of 5704 unique stellar objects contained within the SOPHIE archive. Finally the function ***alpha_abs()*** alphabetises the remaining list for ease of use when reading.

9. SOPHIE Archive Population Distribution

9.1 SIMBAD

Since the strong absorption features we are looking for are anomalous for active main sequence stars, we further want to identify these stars from the archive. It was therefore useful to know what the population distribution of spectral types and magnitudes were in the SOPHIE archive.

Although this information is contained within the spectral image files of each star in the archive, it would necessitate that each file would be downloaded and this information extracted. Given the huge number and the large file size of these high quality spectra it would not be practical to download them all just to extract this information. Fortunately the online astronomical database SIMBAD (*http://simbad.u-strasbg.fr/simbad/*) contains information for millions of astronomical objects, and can perform cross-identifications for lists of such objects. By submitting the list of SOPHIE archive star identifiers created previously, the desired information for these stars was displayed in a generated table.

Some of the identifier names used by the SOPHIE archive were not recognised by the SIMBAD search algorithm. This reduced the list of stars for which information could be obtained to 5145. The number of stars which were not recognised was large enough that it would not be practical to find correct identifiers by other means, but not significant enough to warrant concern for the purposes of this project.

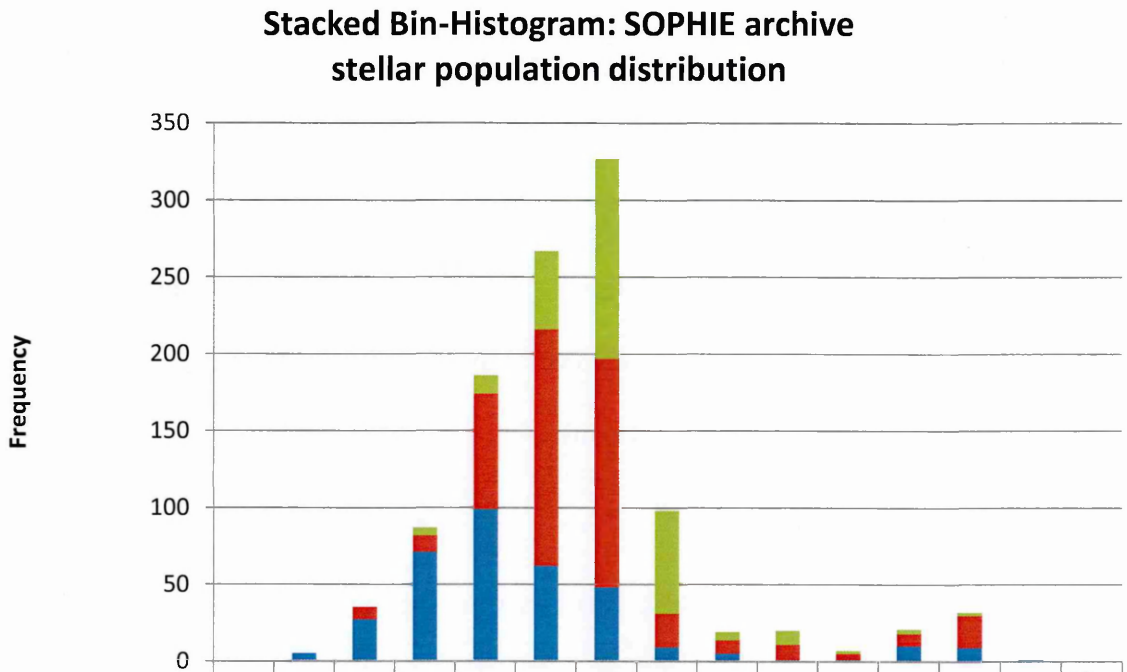
9.2 Excel Data management

The table generated by the SIMBAD search contained the information of the identifiers, magnitude and spectral types for each of the 5145 stars in the query list. To manage this extensive data this table was imported to a spreadsheet in EXCEL.

From this table we wanted to identify main sequence stars of spectral types F, G and K. These parameters best represent main sequence stars which would show the strong emission features expected if intermediate material was not present. To achieve this the cells of a new column were programed with the code as follows:

```
=IF(AND(ISNUMBER(SEARCH("F",E4)),ISNUMBER(SEARCH("V",E4)),NOT(ISNUMBER(SEARCH("I",E4)))),"FV",IF(AND(ISNUMBER(SEARCH("G",E4)),ISNUMBER(SEARCH("V",E4)),NOT(ISNUMBER(SEARCH("I",E4)))),"GV",IF(AND(ISNUMBER(SEARCH("K",E4)),ISNUMBER(SEARCH("V",E4)),NOT(ISNUMBER(SEARCH("I",E4)))),"KV",""))))
```

where the column **E** highlighted in the above code is the spectral type column. This code uses Boolean logic statements to display a confirmation statement when information matching these criteria was found in the corresponding cells of the spectral type column. A small sample region of the vast SOPHIE catalogue data table is shown in Table 2 where it can be seen that this column was working correctly. Using EXCEL's Advanced Filter tool the data subset identified by this programed column was then used to generate the binned magnitude histogram shown in Figure 13. A frequency table containing the exact number of instances for each spectral type shown is shown below the histogram.

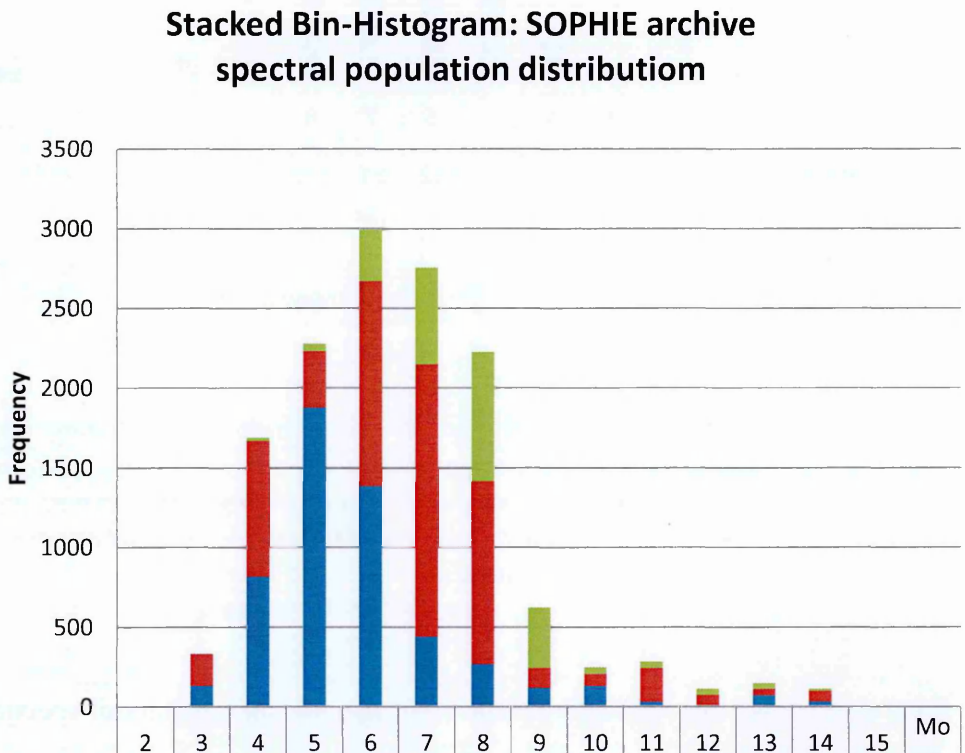


	2	3	4	5	6	7	8	9	10	11	12	13	14	15	More
Frequency (KV)	0	0	0	5	12	51	130	67	5	9	2	3	2	0	0
Frequency (GV)	0	0	8	11	75	154	149	22	9	10	4	8	21	0	0
Frequency (FV)	0	5	27	71	99	62	48	9	5	1	1	10	9	1	0

Figure 13: SOPHIE FV, GV, KV stellar population: binned magnitude histogram showing the frequencies of each spectral type stacked in each column. Below the chart is a frequency table showing the data in the accompanying histogram in numerical form. The stars have been categorised into 1 magnitude wide bins where the stars in each column have magnitudes: $m_V \geq$ column no.

Figure 13 shows the stacked frequencies of main sequence stars of spectral types, F, G and K, contained within the SOPHIE archive, given in bins of 1 mag. Of the 5145 unique stellar objects recorded in the SOPHIE archive, 1105 are main sequence stars of spectral type F, G and K, with the majority of these stars brighter than magnitude 9. This is amenable to our investigations as bright candidates yield higher quality spectra essential for the precise measurement and study of this phenomenon decreasing the effects of background noise dominating the low flux regions of interest.

It is also useful to know what proportion of the 43,624 spectra contained in the SOPHIE archive are from these main sequence F, G and K stars. Using the information in the SOPHIE catalogue data table (a small region of this table is shown in Table 2) the spectra of just stars were identified and extracted from the full archive list. This resulted in a list of 13,786 spectra, for the F, G and K main sequence stars in the SOPHIE archive. Using the same steps described above this list of spectra was used to create the stacked bin-histogram shown in Figure 14.



	2	3	4	5	6	7	8	9	10	11	12	13	14	15	More
Frequency (KV)	0	0	18	45	323	604	808	379	42	41	36	36	12	0	0
Frequency (GV)	0	200	853	355	1286	1711	1151	126	75	211	72	40	67	1	0
Frequency (FV)	0	129	814	1874	1382	437	264	115	127	29	1	69	31	1	0

Figure 14: SOPHIE FV, GV, KV spectra population: binned magnitude histogram showing the number of spectra for each spectral type stacked in each column. Below the chart is a frequency table showing the data in the accompanying histogram in numerical form. The stars have been categorised into 1 magnitude wide bins where the stars in each column have magnitudes: $m_V \geq$ column no.

As mentioned previously we wanted to test the effectiveness of the data reduction program using a select sample of targets. To test this program's effectiveness for the variety of parameters presented in these archives, the targets were to be chosen across ranges of spectral type and magnitude. Additionally as the phenomenon we seek to study arises in planet hosting stars some of targets chosen should be known planet hosts.

To identify planet hosting stars a list was obtained from the online exoplanet database exoplanets.eu in a similar fashion to that of the SOPHIE archive. This list was then put through the 'starfinder.py' program to create a list of 1221 unique planet hosting stars. These stars were then identified in the SIMBAD database simply to ensure that the same star identifiers were being used as for the SOPHIE archive. Again some of the identifiers used by exoplanets.eu were not recognised in the SIMBAD database reducing the number of planet hosts to 1118. This table was likewise imported to Excel and a new column of cells was programed using the code as follows:

```
=IF(SUMPRODUCT(COUNTIF(B4,"*"&$J$4:$J$1121&"*"))>0,"Planet Host","")
```

This code checks whether the star identifier given in the corresponding cell of the star identifier column exists in the table of known planet hosts. If this criterion is met then the statement "Planet Host" will be displayed in the cell of the far right hand column. This can be seen in the region of the data table shown in Table 2 below.

Identifier	Mag V	Sp type	F,G,K	Planet Host?
2MASS J06425417-0120377	15.15	F8V	FV	
CoRoT-13	15.039	G0V	GV	Planet Host
Cl* NGC 2682 YBP 1137	14.92	G4V	GV	
Kepler-420	14.867	G5V	GV	
Cl* NGC 2682 YBP 1392	14.81	G4V	GV	
CoRoT-8	14.797	K1V	KV	Planet Host
Cl* NGC 2682 YBP 1514	14.78	G5V	GV	
V* V354 Mon	14.708	K4V	KV	
CoRoT-6	13.8	F5V	FV	Planet Host
Cl* NGC 2682 YBP 1303	14.64	G3V	GV	
Cl* NGC 2682 YBP 1194	14.61	G5V	GV	
Cl* NGC 2682 FBC 3099	14.55	G3V	GV	
CoRoT-1	13.6	G0V	GV	Planet Host

Table 2: small sample region of SOPHIE catalogue data table

Using this column a minimum of 225 planet hosting stars were identified to exist within the SOPHIE archive. Using the information from this table 24 target stars were selected, 12 stars with known planets and 12 stars with no known planets. These targets' spectra will be used to test the background subtraction. The stars chosen cover a variety of spectral types (F, G and K) and magnitudes (0-15). Information for these targets is shown below in Table 3.

Identifier	Mag V	Sp type	F,G,K	Planet Hosts?	SOPHIE Archive identifier	Obs. Night (highest SNR)	sn26
* ups And	4.1	F9V	FV	Planet Host	HD009826	13/10/2006	610
TYC 3727-1064-1	9.85	F5V	FV	Planet Host	TYC3727-1064-1	28/01/2008	66
HD 150706	7.029	G3V	GV	Planet Host	HD150706	27/02/2009	151
HD 89307	7.03	G0V	GV	Planet Host	HD089307	27/02/2009	163
BD+31 2290	8.96	K4V	KV	Planet Host	BD+312290	07/04/2007	57
TYC 3239-992-1	10.47	F8V	FV	Planet Host	TYC3239-0992-1	28/09/2008	86
BD+47 2936	9.47	K4V	KV	Planet Host	BD+472936	30/11/2011	93
BD+41 3306	8.86	K0V	KV	Planet Host	BD+413306	20/09/2007	57
HD 109246	8.76	G0V	GV	Planet Host	HD109246	29/03/2008	57
V* HN Peg	5.95	G0V	GV	Planet Host	HD209458	16/10/2006	189
HD 20367	6.41	F8V	FV	Planet Host	HD020367	27/10/2009	165
V* V376 Peg	7.63	G0V	GV	Planet Host	HD206860	25/09/2013	200
* bet Com	4.25	F9.5V	FV		HD114710	11/03/2007	463
* eta Cas	3.44	G3V	GV		HD004614	21/02/2008	673
HD 23375	8.58	F0V	FV		HD023375	30/11/2006	92
* tet UMa	3.18	F7V	FV		HD082328	20/12/2006	672
HD 96692	9.8	K5+V	KV		HD096692	03/02/2007	81
HR 3901	6.42	F8V	FV		HD085380	05/03/2010	92
HD 25457	5.38	F6V	FV		HD025457	26/09/2013	138
BD+63 499	9.86	K2V	KV		BD+630499	01/12/2010	50
HD 38847	8.84	G0V	GV		HD038847	27/11/2007	53
HD 182274	7.8	F6V	FV		HD182274	20/09/2007	60
HD 156826	6.32	G9V	GV		HD156826	14/04/2008	98
* 61 UMa	5.34	G8V	GV		HD101501	10/06/2007	219

Table 3: 24 candidate stars, chosen over a range of magnitudes (0-15) and spectral types (F, G, K)

The data for these candidates was then downloaded from the SOPHIE archive. Since we are performing our own data reduction, and not using the archives own automated process, we required the raw unprocessed data for each star. This requires that an entire nights observing data be downloaded.

To find the correct nights the archive was searched for each target, returning a tabulated list of all of the nights for which that target has been observed. This list provides information of the observation mode used (HE or HR, see Section 5.2) as well as the highest achieved signal-to-noise ratio (SNR) for each night, as read from the FITS file headers (see Section 10). In each case the night with the highest SNR using the HR mode was chosen and downloaded.

10. CCD Processing and Preparation for ECHOMOP

Background CCD Échelle data reduction

The different systematic and noise contributions that arise are specific to the instrument being used. To fully account for these effects it is therefore important to understand the instrumentation used to collect the data in order to differentiate between signal and noise. As with many astronomical instruments, the detector used by the SOPHIE spectrograph is a CCD chip. Systematic noise contributions specific to CCDs are accounted for by taking specific calibration image frames which can be used to isolate and remove these unwanted effects.

Read out noise, introduced by the electronics of the image sensor, can be accounted for by taking a Bias frame. Their purpose is to capture and subtract the DC offset, which is always present and fairly constant in every image frame. Bias frames are short exposures taken with no light falling on the image sensor. This will capture the readout values produced by each unexposed pixels on the CCD. The readout value recorded by the 'empty' unexposed pixels will have a small distribution around zero, meaning that negative values could be recorded in the output image. To avoid negative values, the CCD electronics are set-up to introduce a constant positive offset level to the readout signal. This sets a new mean 'zero level' to the image called the Bias level. The Bias frames will thereby capture this Bias level of the CCD instrumentation as well as the associated noise background. Typically multiple of bias frames are taken for a given session, which are stacked and averaged to create a master Bias frame.

Also needing consideration is variation in the sensitivities between pixels on the chip. This causes the pixels to not be in agreement when recording light of the same intensity. To account for this 'flat field' frames are taken. These are exposures of a bright near uniform light source, usually attained using a suitable continuum lamp (typically

tungsten). The idea here being that should all pixel sensitivities be identical then the same value should be universally recorded. This not being the case reveals and allows the discrepant variation between pixels to be normalised. As with the Bias frames, multiple flat field frames are taken and stacked and averaged to form a master Flat field frame.

During image processing the master Bias frame will be subtracted from every science frame as well as the master Flat field frame, thereby removing the offset Bias level as well as the read noise. Then the Science frame (Bias subtracted) is divided by the master Flat field frame (Bias subtracted) to give the resulting reduced output image. In equation form:

$$Final\ Reduced\ Science\ Frame = \frac{Raw\ Science\ Frame - Bias\ Frame}{Flat\ Field\ Frame - Bias\ Frame}$$

When recording spectra using a CCD, there is an additional need to calibrate the recorded image for wavelength. This is done using 'arc' frames. These are exposures where a calibration light source, with known identifiable spectral lines, has gone through the instrument just as the light from the distant target would. In this way the constituent wavelengths will be dispersed to the same extent as the target light and by matching where these wavelengths fall on the CCD the target spectra wavelengths can be identified.

Additional factors need to be taken into consideration when handling Échelle CCD images. The spectra of these instruments appear as a series of stacked orders on the chip. These orders have a more complex shape and distribution than for a single spectrum increasing in curvature and separation with increasing wavelength, as can be seen in

Figure 10. This more complex arrangement of the data on the image must be accounted for to get the data to a usable form. This is achieved using dedicated Échelle data reduction software packages.

Raw night's data

The raw data which can be downloaded from the SOPHIE archive for any given night comes in the form of a compressed (.tar) file which needs to be extracted before the data can be worked on. The extracted folder contains the raw image files, in FITS (Flexible Image Transport System) format, for each exposure taken on the night, including the bias, flat field, arc and object frames. FITS (.fits) files are the most commonly used format for astronomical image data storage. The FITS image format was designed primarily to store scientific data sets. FITS files can be comprised of multiple segments called Header/Data Units (HDUs), which can store multidimensional data arrays. Each HDU consists of an ASCII formatted 'header unit' which provides descriptive information of the accompanying 'data unit'.

The SOPHIE data have a naming convention in which all files in a raw night's data folder are named for their respective times of creation and give no indication of their contents or exposure type. This information must be obtained from the FITS 'header' for each image.

Need to prepare frames for ECHOMOP

For this work the Starlink (*http://www.starlink.ac.uk/echomop*) Échelle data reduction package ECHOMOP was used. Before the CCD images could be used by ECHOMOP they had to be processed. Scripts can be written in the programing language cshell which

call Starlink commands to prepare suitable input files for the ECHOMOP package from the raw SOPHIE archive data.

Pipeline overview

Fortunately such scripts have already been written for similar Échelle data reduction projects which employed ECHOMOP. The scripts prepbias, prepflat, prepobjs, preparcs and preprun_ext1_auto were written by Andrew Collier Cameron (University of St. Andrews) and Martin Clayton (UCL/Starlink) to prepare the bias frames, flat-field frames, object frames, arc frames and to manage these four scripts respectively. To get the raw data into a form that can be used by these scripts the additional scripts make_ndfs, makelog and preprun_master were written by John Barnes (The Open University). After a few adjustments these scripts were made to work with SOPHIE data as well as with my computational setup. The function and operation of each of these scripts are explained in order of execution below, and in full detail in the appendices.

STAGE 1: Frame preparation

Makelog – creates a log file, sophie.log, containing information of each files contents as read from their FITS headers. This will be referenced by subsequent scripts to inform file treatment. (Appendices 3.1)

Make ndfs – this script converts image files to Starlink's native NDF format required for Starlink operation. (Appendices 3.2)

Preprun master – sorts files by type using sophie.log in preparation for subsequent ECHOMOP preparation scripts. This script then initialises the script 'preprun_ext1_auto'. (Appendices 3.3)

STAGE 2: ECHOMOP preparation

Preprun_ext 1_ auto – This script runs all of the frame preparation scripts for ECHOMOP (prepbias, prepflat, prepobjs, preparcs). The script first establishes the

CCD parameters of the overscan and clipping regions, the gain and the read-out noise required for the subsequent scripts. (Appendices 4.1)

Prepbias – script produces a single median image from a series of raw CCD bias frames. (Appendices 4.2)

Prepflat – script produces a single median flat field image from a series of raw flat field frames. This image will be used as the single flat field frame in ECHOMOP. (Appendices 4.3)

Prepobjs – this script performs the basic CCD data processing steps of bias subtraction and trimming of the object images making use of data produced by the previous scripts preprun_ext1_auto and prepbias. (Appendices 4.4)

Preparcs – this script performs the similar CCD processing steps described in prepobjs, on the calibration lamp frames preparing them for use for wavelength calibration steps. (Appendices 4.5)

After all of these scripts have been run for a given night's data folder all of the fully prepared frames necessary for use in ECHOMOP's data reduction process will have been created and stored in a subfolder titled 'Reduce'.

Tweek_pipeline

In order for the ECHOMOP preparation scripts to be run on a night's data, copies must exist in the same working directory as the raw data files. Additionally, information of the date particular to the night of observation has to be changed within the scripts themselves. These do not present an issue for a small number of cases, but given the greater number of target spectra used in this work and with the future promise of large scale archival reductions more automation would be desired. To this end the script tweek_pipeline was designed, written in cshell to automate this process (Appendix 5). This script manages the execution of the ECHOMOP preparation scripts, described above, for any number of extracted nights' data folders in a given directory. For this script to run it must exist within the same directory as the archival data folders.

The date information needs to be manually edited into the ECHOMOP preparation scripts when they are run for the raw data of any given night. The tweek_pipeline script removes the need to edit this information within the preparation scripts by instead reading the date information from the nights' data folder names to be used as an input argument. The script has been designed to work on the raw nights' data folders as they come from the SOPHIE archive. It may be the case that the archival data folders are stored within a directory also containing unrelated files and folders that we do not want to process. To differentiate between these folder types the script will only run for folders that have been prefixed with a specific keyword. Currently this keyword is 'T0' but it can be edited within the tweek_pipeline script by the user as desired. Therefore folders that are intended to be part of this process should be renamed with the prefix 'T0' and then some numerical or alphabetical characters should a particular execution order be desired (e.g. T01, T02, T0A etc.). Otherwise this key word can be amended to the word 'SOPHIE' removing the need to rename folders provided that only archival data folders exist in the working directory. Once these preparatory steps have taken place and the script executed, the script will perform the following processes for each:

- Changes current working directory to T0.../raw/'date' following the structure convention of the archival folders.
- Records the date information from the directory title in a form which can be used by upcoming scripts.
- Copies all cshell ECHOMOP preparation scripts into the current working directory
- Runs the scripts makelog, make_ndfs, and preprun_master, with the latter running the remaining preparation scripts.
- Additionally outputs the start time and end time for when the script began and

concluded operations on each night's folder.

Upon completion of this script's operation, the folders of all observation nights contained within the directory will have been fully processed in accordance with the preparation scripts described above. A further advantage of this automation is that since the manual input required to set up and repeat for each data folder has been removed, the inherent lengthy execution times and computational demand can be moved to more convenient hours.

Identifying target object frames

Because each night can have multiple targets, multiple processed object frames can be produced by these scripts. The intended target object frames must therefore be identified by referencing the log file sophie.logHR so that they can be used in ECHOMOP.

Bclean cosmic ray removal

As the data in the object frame exposures has not yet been processed in any way there is a high likelihood they will be subject to contamination from cosmic rays. To ameliorate these features should they be observed, the Starlink function bclean uses statistical analysis to identify and remove the outlying contamination effects. This must be done to any object frames prior to their use in ECHOMOP. This step can be repeated until the user is satisfied with the quality of the cosmic ray removal.

11. ECHOMOP

ECHOMOP is a Linux based software package which provides a set of tasks to efficiently manage reduction of Échelle spectra. The package may be used in a variety of ways depending on the experience of the user and the desired image analysis. The many different options of the package allow for different levels of automation and extent of reduction. The package does not have the facility to perform CCD processing which must be done prior to implementation. This we have already done with the cshell preparation scripts as outlined in the previous Section.

ECHMENU

The most convenient way of using the ECHOMOP software, and the way it was used for this work, is to use the main ECHMENU task. This provides a guided path through the standard reduction processes. For the purposes of our work, we are so far only concerned with achieving a good background subtraction, not requiring the full extent of the reduction options available. The options used were as follows:

Option	**Purpose**
1:ech_locate	order location
2: ech_trace	order tracing
3:ech_fitord	order polynomial fitting
4:ech_spatial	slit definition, determining Dekker/object limits
5:ech_ffield	flat-fielding
6:ech_sky	models sky background
-or-	
22:ech_mdlbck	models scattered light
7:ech_profile	models object profile
8:ech_extrct	extracts object and arc spectra
14:ech_result	writes results to file

The full breakdown of each stage of the reduction process including all input parameters and data frames used is given in Appendix 6. Other options can continue on to perform

wavelength calibration and blaze correction although these were deemed not necessary for the purposes of this project.

Background subtraction

ECHOMOP provides two options for background characterisation and subtraction, Option 6:ech_sky and Option 22:ech_mdlbck each taking different approaches. In Option 4 the Dekker is determined, which defines the object limits along each order as well as the 'sky' pixel regions on each side. Option 6 is the default subtraction option, it uses the adjacent sky pixels to model the sky intensity beneath each order. In many cases this will be adequate, however, it is susceptible to inter-order contamination. Option 22 can be used in place of Option 6 in cases where there is severe scattered light contamination. This option works by fitting spline-polynomials to the inter-order regions (the gaps between order Dekker limits), to produce single global surface to the background. This approach is much more computationally taxing than Option 6 and is usually avoided unless necessary. Since our investigations concern wavelength regions where the flux is typically low, the accuracy of the background subtraction is essential. As the ECHOMOP background subtraction options are part of established software we are unable to see how the underlying code is manipulating the data at every stage. It is also unclear how accurate the background subtractions being achieved are. Most investigations are not concerned with low level flux, and so the accuracy of the background subtraction is not as important as it is to our investigations. For this reason there was concern that the ECHOMOP background subtraction options may not be accurate enough for our investigations. The effectiveness of these options is investigated in the next Section (Section 12 Background subtraction).

Reduction database file

For each reduction an associated file known as the reduction data base file is created in Starlink NDF format (file extension .sdf). This file keeps all the housekeeping data for a particular reduction in one place including all sets of data frames associated with the reduction. Using this file provides the ability to replay any aspect of the reduction at a later date.

Output

When the reduction process has been completed the end result is a collapsed Échelle spectrum. Here all curvature, separation and non-signal contributions have been removed from the orders, leaving just the spectral order data stacked in a single image. Figure 15 (below) illustrates this showing an unprocessed Échelle CCD image before the reduction (top panel) and the resulting collapsed Échelle spectrum after the reduction (bottom panel). The data shown in this collapsed spectrum is the spectral order data stacked in one pixel high rows. Storing the data in this form allows for the easy extraction of spectral order data for analysis. This collapsed spectrum is the final output image as saved by Option 14, in NDF format.

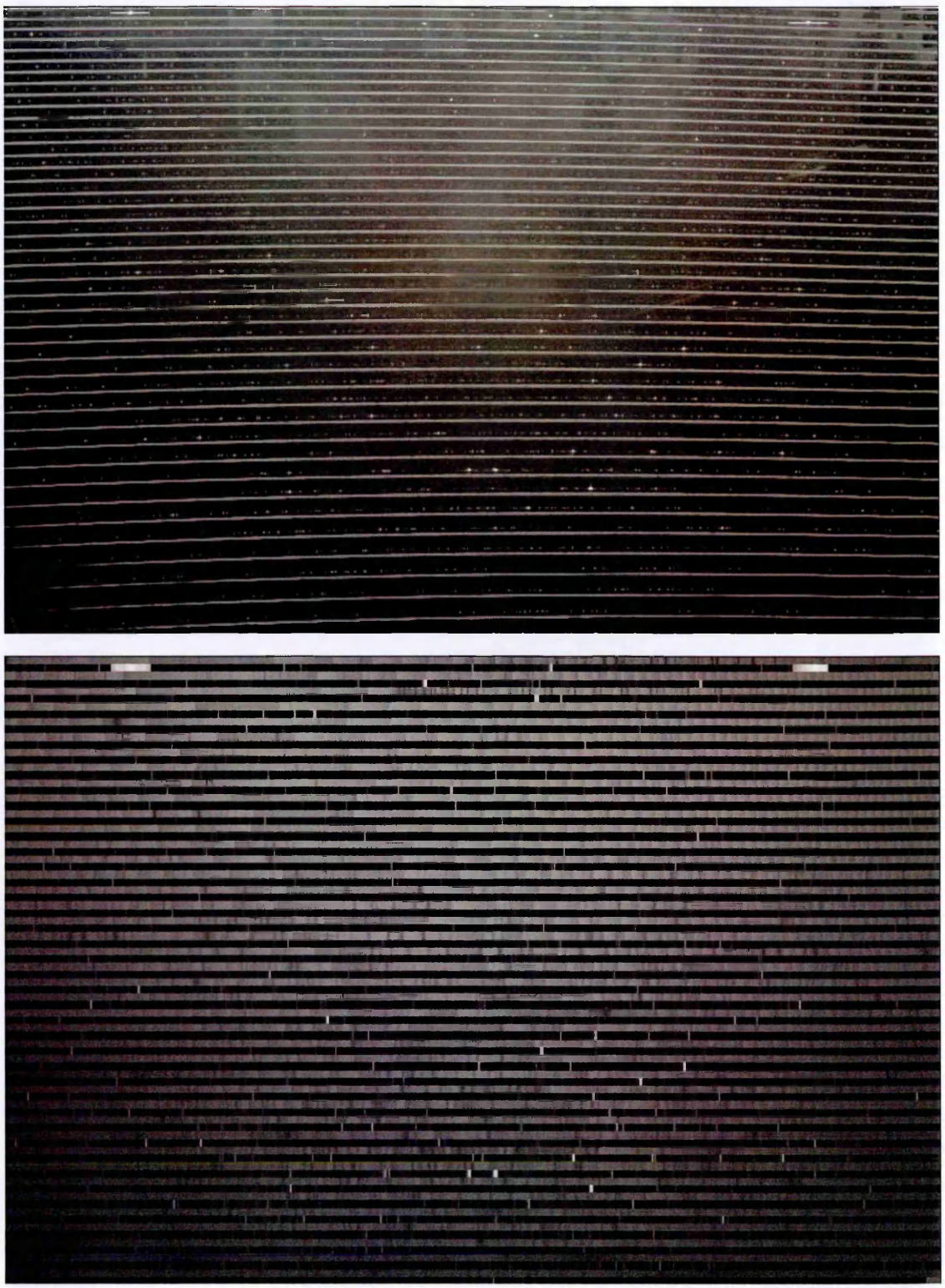


Figure 15: The Échelle spectra of star ups And (HD 9826) shown before (top panel) and after (bottom panel) the ECHOMOP reduction process

12. Background Subtraction Software, SKYBM

All instrumental conditions being ideal, the only signal being measured by the CCD would be the object spectral orders and the background should be essentially zero. This means that any non-zero background will be the result of light contamination. As discussed previously the planetary mass loss phenomenon we are investigating causes extreme absorption in the Ca II H K line cores. The flux in the regions centred on these lines is always low due to stellar photospheric absorption as well as a further extinction form the Earth's atmosphere. This means that the signal in these regions is harder to discern from background noise sources. For this reason the background noise contributions present a significant obstacle and have a much larger impact on the data than would be the case for a more typical wavelength regime in a stellar spectrum. It is therefore of great importance to characterise and remove this underlying contribution in order to accurately study the stellar spectra in this region.

The ECHOMOP pipeline includes its own background subtraction protocols, which for the reasons discussed in Section 6, may not be suitable for our investigations. Since an accurate background subtraction is so key to this work it was decided that we should write our own background subtraction software in Python. This was to ensure that we were familiar with the 'nuts and bolts' of how the data was being manipulated and also a chance for to improve programing skills.

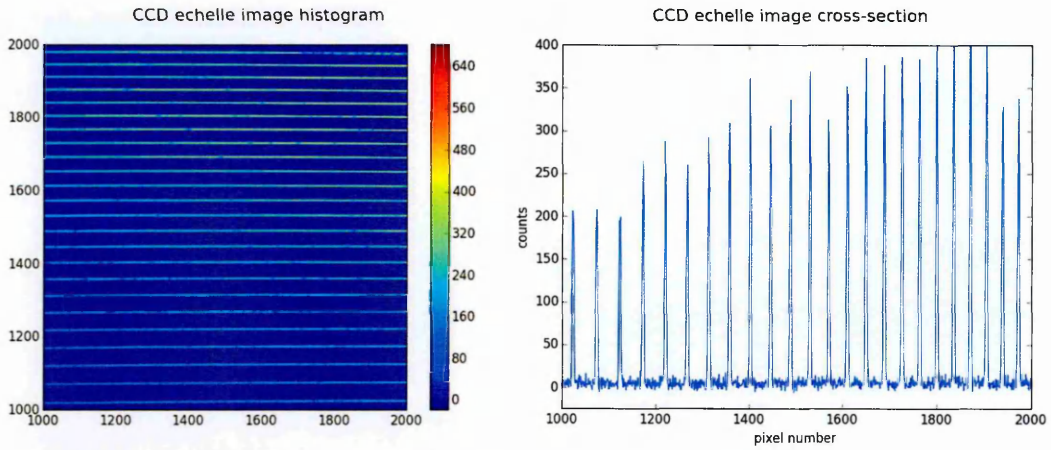


Figure 16: small region of CCD Échelle image data. Left panel: 2D image histogram, right panel: plot of vertical cross-section

Figure 16 shows a small region of an Échelle spectrum, as displayed using matpoltlib in Python. The left hand panel shows a false colour image histogram of the data array where the light blue regions are the Échelle orders from the target light and the dark blue regions are the background. The right hand panel shows a vertical cross-section of this data where the Échelle orders can be seen as clear peaks.

The background subtraction software works by first creating a mask to distinguish between the data and the background in the CCD image. This mask is then used to cover the data regions isolating the background so a shape can be fitted to this background estimating the contribution of light contamination which can then be removed from the image. A key objective of this program was to provide a high degree of user interactivity to ensure the process could be tailored for the best outcome.

Illustrated walkthrough of software operation

The software consists of the three scripts, thresh_find_v1707.py, make_mask_vM1707.py and SKYBM.py which must be executed in this order.

Additionally the script SKYBM_functions_v1707.py contains a library of functions designed for this software and utilised by all three scripts. The full code for each of these scripts is given in Appendix 7.

What follows is an illustrated walkthrough of the execution of these three scripts using the test Échelle spectrum image shown in Figure 17.

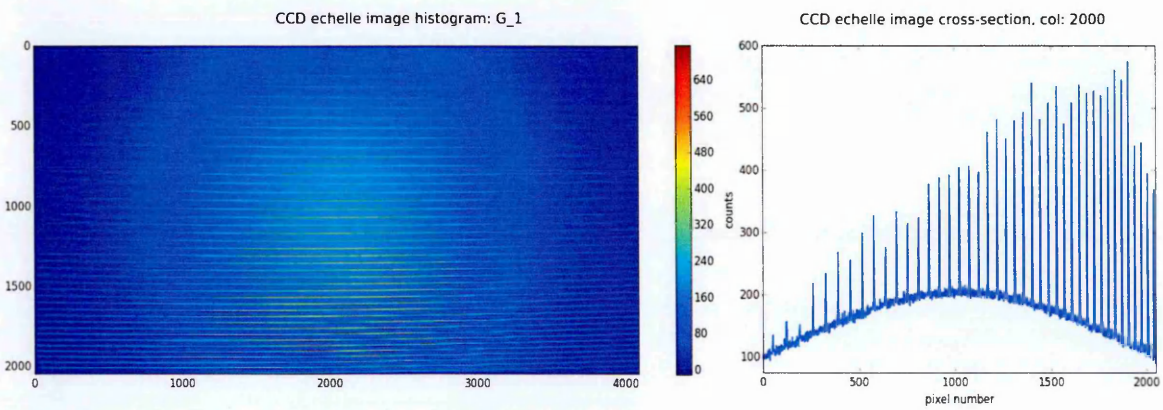


Figure 17: Gaussian perturbed Échelle spectra, unprocessed CCD image. Left panel: full 2D image histogram, right panel: cross-section at col. 2000.

For the sake of demonstration the Échelle image used has been artificially perturbed with a Gaussian shape to the background, to represent the unwanted background light contamination. This exaggerated background shape is not representative of normal light contamination, but instead serves to illustrate the working of the background subtraction. This software should be applied to the object frames before ECHOMOP and after the CCD processing scripts and bclean cosmic ray removal. These scripts require that all files used be in FITS format, therefore they must be converted from NDF format before these scripts are used and then converted back for use in ECHOMOP. When using the resulting background subtracted frames in ECHOMOP neither Option 6 or 22 need be run. However, if the ECHOMOP options were to be run on top of the SKYBM

subtraction, there would be very minimal difference in the output as the background has already been subtracted.

12.1 thresh_find_v1707.py

The first step in this procedure is to find a suitable 'threshold counts value'. This is an integer value that will be used by the mask making procedure to define the boundary between background and data regions. This script allows the user to test guesses at this value and should be rerun until an appropriate value has been found.

First the user is prompted for how many 'cuts' of their chosen object image file they would like to check. This is the number of evenly spaced vertical image array columns (image cross sections perpendicular to Échelle orders, see Figure 16, they would like to display. In doing so the

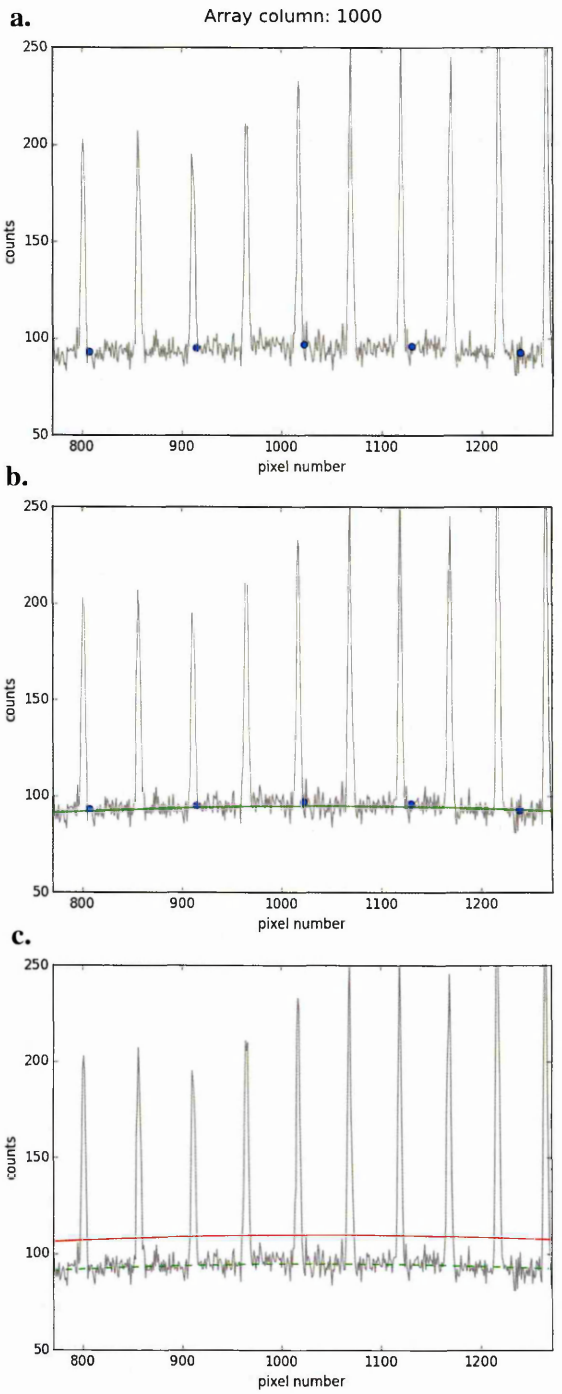


Figure 18: plots to demonstrate the operation of the working of the thresh_find script. A region of the data from column 1000 is shown in each plot by the grey line. Panel a: blue circles represent the evenly spaced median binned data points for the data range shown. Panel b: green line shows the interpolated spline fit to the blue data points. Panel c: red line shows the 'threshold line' created by shifting the green line up by the number of counts chosen by the user.

threshold value can be manually inspected ensuring that is appropriate across the whole image. Where the image array is $I[n_y, n_x]$ a cut is $I[:, n_{cut}]$, with : denoting the entire column.

The user is then prompted for an initial guess for the 'threshold counts value'. For each of the given number of $I[n_{cut}, :]$ columns 20 median binned data points are created by calculating the median values of 20 equal sized fractions of the column data. This is shown in Figure 18.a where a region of the data in column: 1000 of the test array (see Figure 17) data is shown by the grey line, upon which the evenly spaced median binned data points have been fitted, represented by the blue circles. A spline is then fitted to these data points thereby creating a rough estimate of the background, shown as the green line in Figure 18.b. This line is then shifted up by the given threshold counts value thereby forming the 'threshold line' which is the boundary definition between the background and data, shown as the red line in Figure 18.c.

This procedure will be performed for every column in the image by the mask making procedure to generate the mask. Plots of each of the evenly spaced cuts are produced showing the column data in grey and the threshold line in red for contrast. Two resulting plots of three cuts made to our test image are shown in Figure 19.

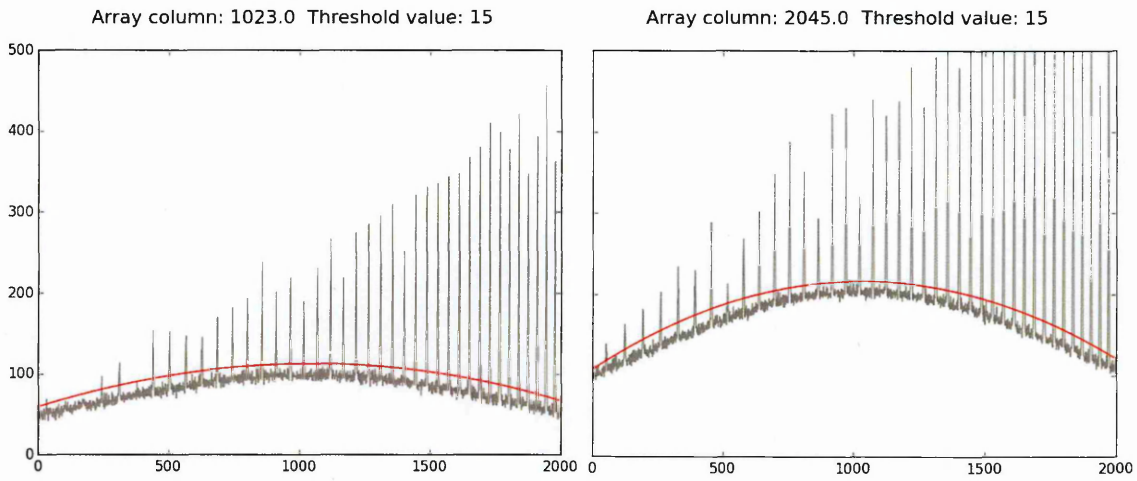


Figure 19: Two cross-section plots showing the image column data (grey) and the threshold line (red). left: col 1023, right: col 2045

12.2 Make_mask_vM1707.py

The next step is to create the mask image. Using the 'threshold counts value' found using 'thresh_find_v1707.py' this script will generate a threshold line for each column in the image in the same way. For each column these respective lines will be used to distinguish between the data and the background. As it does so, an empty array of equal size to the image will be populated with the values 0 and 100 corresponding to the background and data regions respectively. Hence the value 100 corresponds to the regions to be masked. The user is additionally prompted for 'pixel margin value'. This is the number of pixels to increase the widths of the masked regions by if desired to ensure peaks are adequately covered by the mask, see Figure 20.

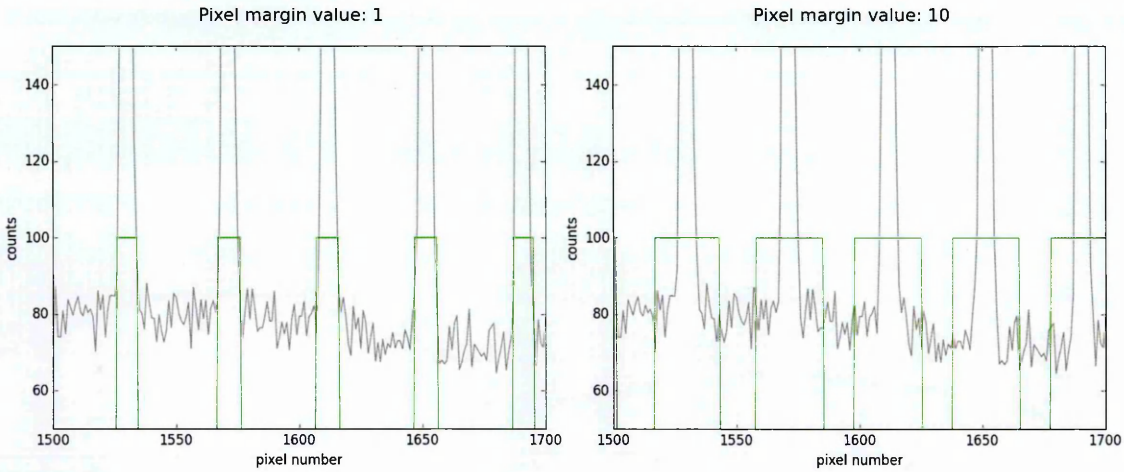


Figure 20: Two plots showing column cross-sections of the same region of the image data (grey) and MASK (green), demonstrating the outcome of different 'pixel margin values' on the resulting MASK. Left panel: pixel margin value = 1, right panel: pixel margin value= 10

To ensure that the orders are sufficiently masked it is recommended, although not essential, that the flat field frame for that night is used in this step instead of the object frame. The bright flat field orders having no spectral lines will ensure that a solid unbroken mask is produced. The resulting mask generated for our test image is shown in Figure 21. Here the object frame was used instead of the flat field.

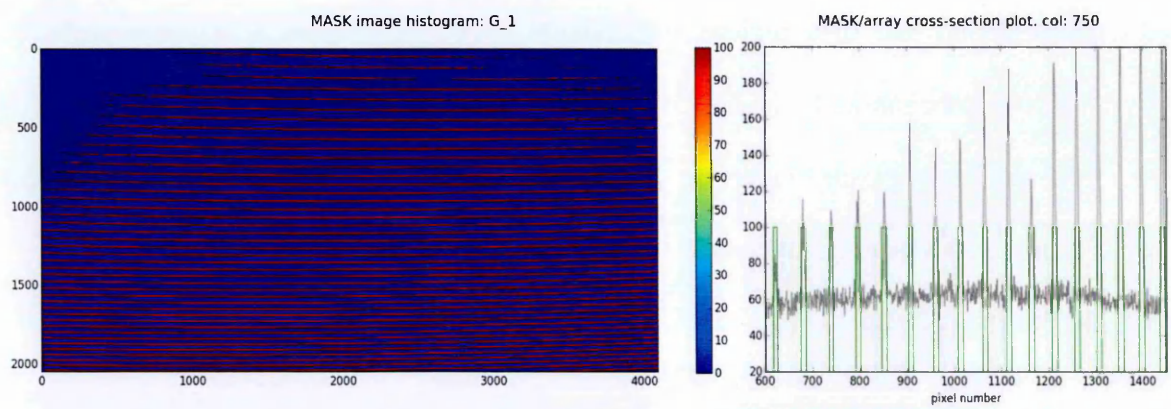


Figure 21: MASK. Left: 2D image histogram, right: data/MASK comparison cross-section col. 750

This mask array is then output to a new FITS file in the current working directory taking the same filename as the input with the suffix '_MASK'.

12.3 SKYBM.py

Finally with the mask produced, the final step of this procedure is to model the background and subtract it from the object image. This script prompts the user for the filenames of the object frame and the corresponding mask. The code proceeds to iterate through the image column by column. For each column median binned data points are created at the centre of each region not covered by the corresponding mask, see Figure 22 (left). An interpolated model line is then fit to these points using a Savitzky-Golay filter (Savitzky & Golay 1964), this is demonstrated in Figure 22 (right).

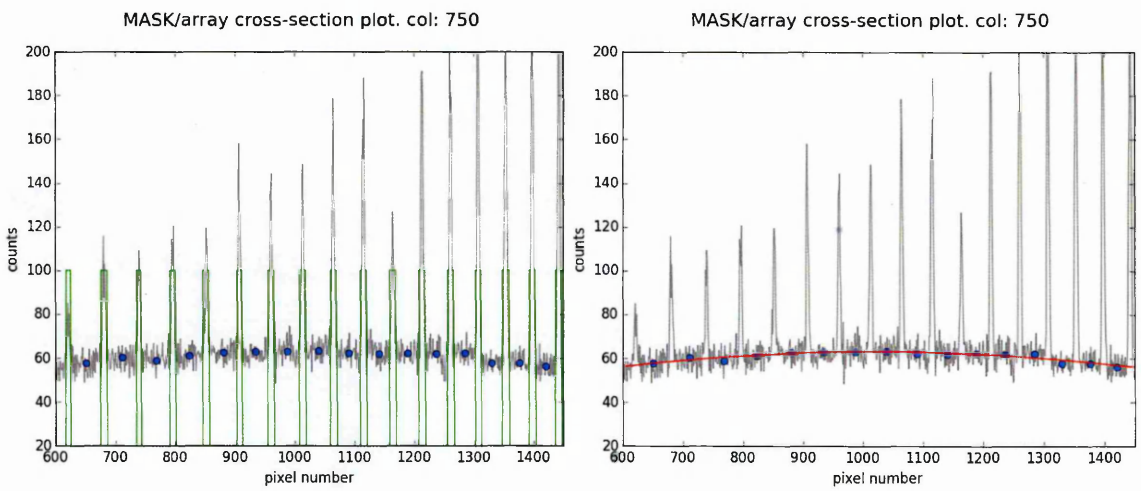


Figure 22: Two cross-section plots demonstrating the background modelling process of the SKYBM software. Left panel: column cross-sections of the same region of the image data (grey) and MASK (green), with median binned data points (blue circles) fitted to the unmasked data regions. Right panel: Interpolated model line (red) fit to the inter-order median binned data points creating the background model for this column.

These modelled columns are stored in a new array and then the same modelling procedure is applied horizontally, resulting in a smooth 2D model of the background

light contribution. This model is then subtracted from the object image array resulting in the final background subtracted image.

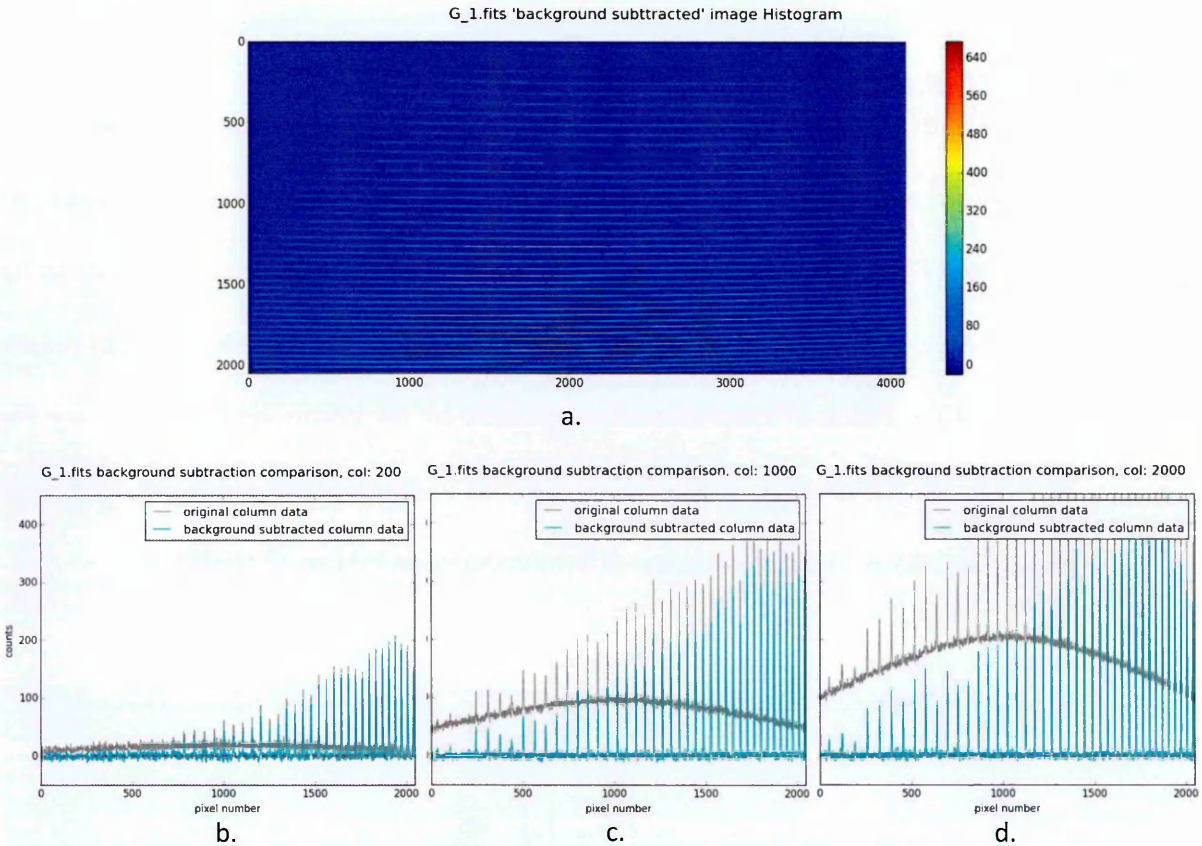


Figure 23: Background subtracted Échelle image data. Panel a: 2D image histogram, panels b, c, d: comparison cross-sections at columns 200, 1000 and 2000. Original background contaminated data (grey), background subtracted data (cyan), zero line (red dashed)

Figure 23.a shows the resulting background subtracted image histogram for our test spectrum. In Figure 23 b, c and d are three cross-sections plots at different locations on the chip, showing the raw data background before (grey) and after (cyan) subtraction, as well as the dashed red line indicating the 0 level for visual evaluation of the subtraction. This background subtracted image is then output to a new FITS file in the current working directory taking the same filename as the object file with the prefix 'BG_'.

This script also provides the option to only use a user-defined subsection of the main image array. This is useful if only a small area of the CCD chip be of interest and cuts down on computational expense for this and all subsequent reduction steps. The Ca II H&K line cores are confined to just two of the Échelle orders on the CCD. For the purposes of this work it is therefore only really necessary to ensure that the background is sufficiently subtracted in these regions.

13. Testing the SKYBM Background Subtraction Software

The aim of this project was to produce a means of reducing raw Échelle spectra from the SOPHIE archive. Vital to this is the ability to achieve an accurate subtraction of the background scattered light especially in the region of the Ca II H and K resonance lines which are the focus of our investigations. In this Section we assess the effectiveness of the SKYBM software background subtraction, as well as comparing the resultant reduced spectra to those using the ECHOMOP background subtraction options.

13.1 SKYBM SUBTRACTION TESTS

A good background subtraction should have the result of reducing the counts measured in the regions between Échelle orders to effectively zero. This is because any counts measured in these regions will be the result of unwanted light contamination as explained in Section 12. To assess the performance of the background subtraction software SKYBM, as described in Section 12.3, it was applied to the Échelle spectra of four test stars, two with high SNR (>100) and two with low SNR (<100).

Star: *eta Cas SNR: 673 (HIGH) observation date: 18:14:56383 21/02/2008

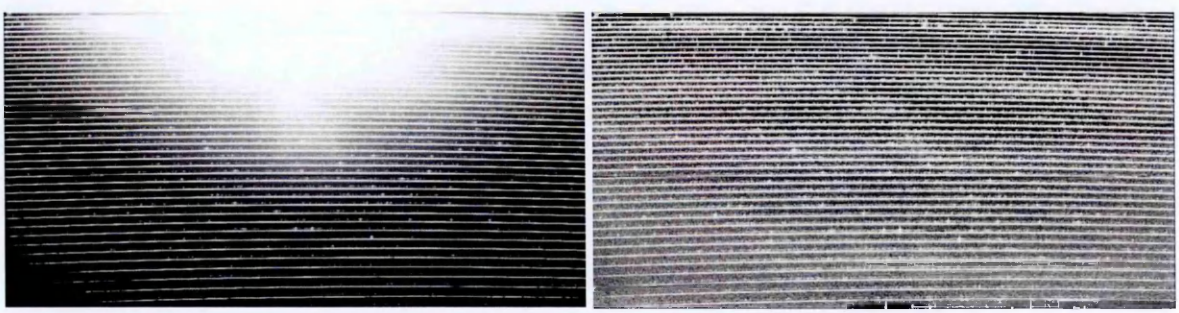


*Figure 24: CCD images of the Échelle spectrum of the star *eta Cas, showing the un-subtracted raw exposure (left) and the background subtracted spectrum (right).*

Figure 24 shows the side-by-side CCD images of the Échelle spectrum of the star *eta Cas (observation date: 18:14:56383 21/02/2008), before and after subtraction using SKYBM software. Both images have been produced by cutting the high value data down to emphasise and reveal the shape of the comparatively low level background. The images are in greyscale to make the differences in intensity in the background more apparent. These images serve as an at a glance comparison, but more quantitative assessment is required to truly assess how well the background subtraction has worked. It can be seen that the before image (left) has a distinct shape to the background scattered light, emanating from the top centre of the chip. Comparatively the after image (right) has a much more even background indicating that it is flatter.

To quantitatively assess the effectiveness of the background subtraction, i.e. that the background is essentially flat at around zero flux, required a more detailed look at the changes in the inter-order regions between the two images. To do this, three pixel rows at different positions on the image were examined. Due to the curvature of the orders it was not possible to examine any entire pixel row that did not pass through an order. The locations of these rows were therefore chosen for containing mostly inter-order data and in three different locations with different background scattered light intensity, as judged by Figure 24 (left). The locations of the Échelle orders are consistent throughout SOPHIE stellar spectra, therefore these same rows were used for each set of comparisons. Figure 25 shows the mask produced for this spectrum where the red regions are the masked orders (set to value of 100) and blue regions are the unmasked background (set to value of 0). The chosen rows are shown as the yellow lines in Figure 25.

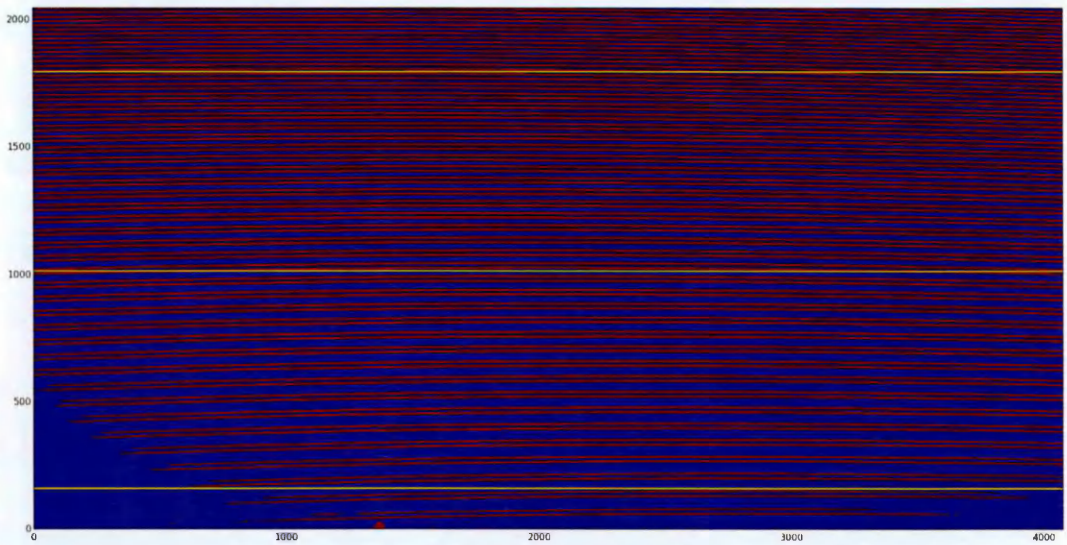
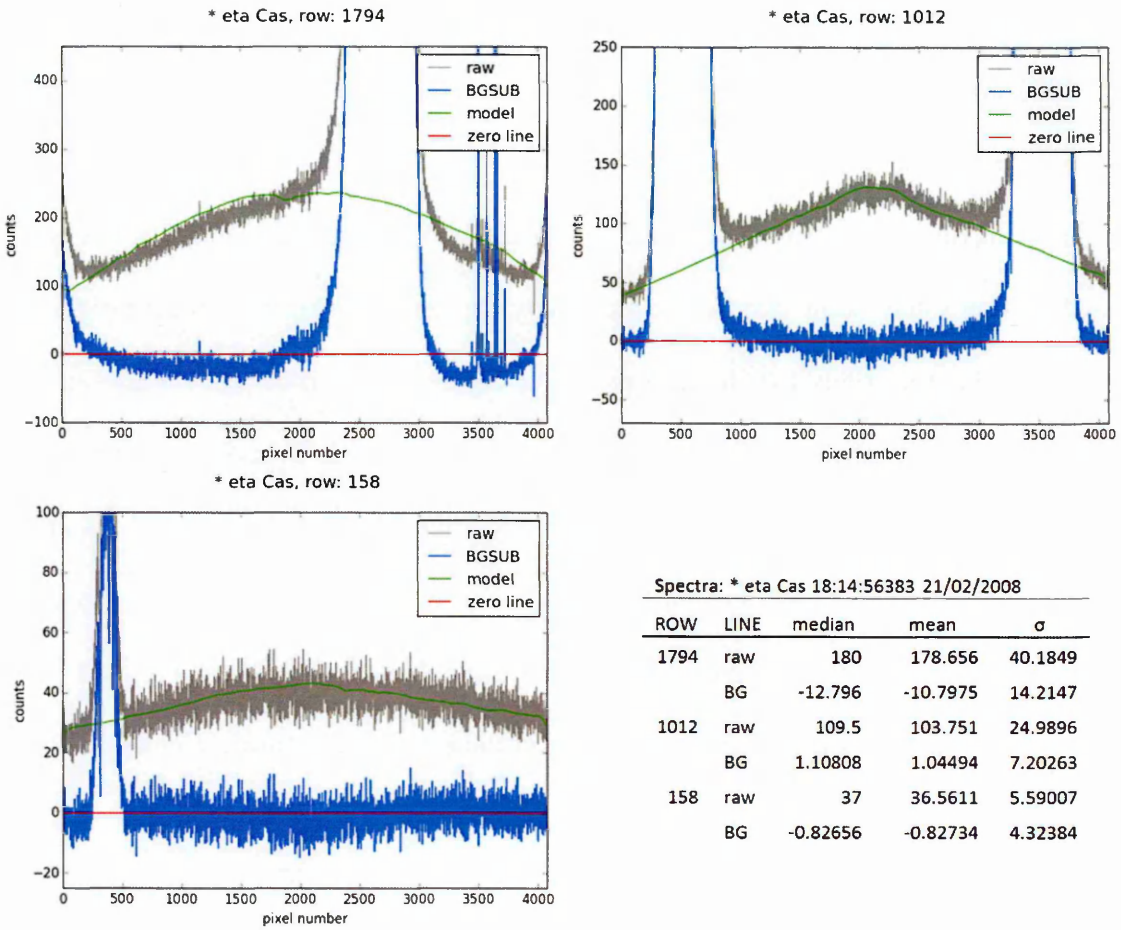


Figure 25: MASK with sub-row locations highlighted i.e. set = 65 (for yellow)

The data from these three rows are shown in the Figure 26 below, where the raw un-subtracted row data is shown by the grey lines, the model fit to the background produced by the SKYBM program is shown by the green lines and the background-subtracted row data is shown by the blue lines. To assess reduction and flatness a red line representing 0 counts is also shown. The plots in Figure 26 are accompanied by a data table giving the mean, median and standard deviation of the background data for each row. The information in this table was obtained by blocking out the Échelle orders using the mask and measuring just the mean, median and standard deviation data of the inter-order background regions.



Spectra: * eta Cas 18:14:56383 21/02/2008

ROW	LINE	median	mean	σ
1794	raw	180	178.656	40.1849
	BG	-12.796	-10.7975	14.2147
1012	raw	109.5	103.751	24.9896
	BG	1.10808	1.04494	7.20263
158	raw	37	36.5611	5.59007
	BG	-0.82656	-0.82734	4.32384

*Figure 26: plots and data table, for the Star: * eta Cas, Date: 18:14:56383 21/02/2008. Top left: row: 1974, Top Right: row: 1012, Bottom left: row: 158, Bottom right: mean, median and std. deviation data table.*

From the plots in Figure 26 it can be seen that the background subtraction has resulted in flatter line profiles at close to 0 counts for each row. However in the case of row 1794, the SKYBM has over subtracted, as reflected by the plot Figure 26 (top left) and shown in the data table (bottom right). This is a result of the SKYBM model not accurately fitting the background. In the row 1794 plot the model (green line) can be seen to over estimate the raw data (grey line profile) therefore causing the BGSUB (blue line profile) to be over subtracted and fall below zero. The poor fit is due to the fact that the Échelle orders in this wavelength region have much smaller separation (see Figure 25). There will therefore be less unmasked background data to fit the model to, resulting in a model

and subsequent subtraction that is not as accurate in this region. Conversely in the rows 1012 and 158 are in a region with much greater spacing between orders and therefore more background data for the model, resulting in much better subtractions as reflected by their median and mean data being much closer to 0. Additionally the low standard deviation values show that the data is closely clustered about the mean demonstrating the data to be essentially flat. In all 3 panels of Figure 26 it can be seen, where an Échelle order occurs, that the background model (green line) made a smooth approximation of the background beneath it. Accurately accounting for the background contribution beneath spectral orders is crucial for the accurate calculation of log R'_{HK}. Judging by the model fit to the background on either side of the Échelle orders, as seen in the lower row plots at least, it would seem that the background is being accurately modelled in these regions.

Star: *ups And SNR: 610 (HIGH) observation date: 00:07:16067 14/10/2006

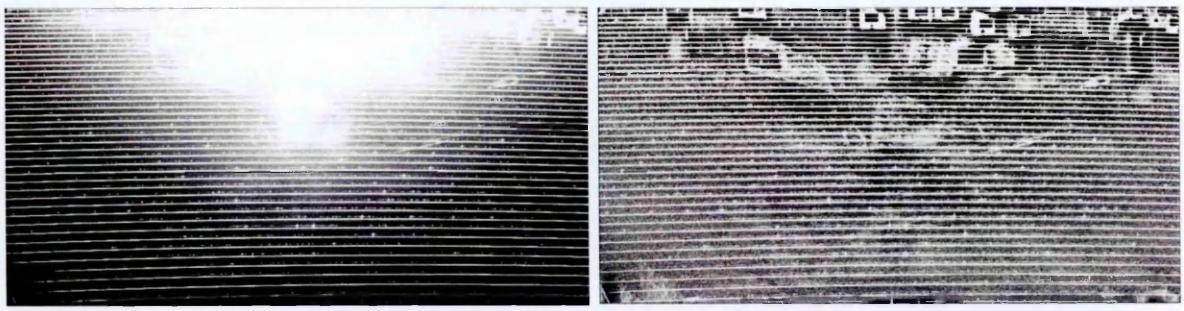
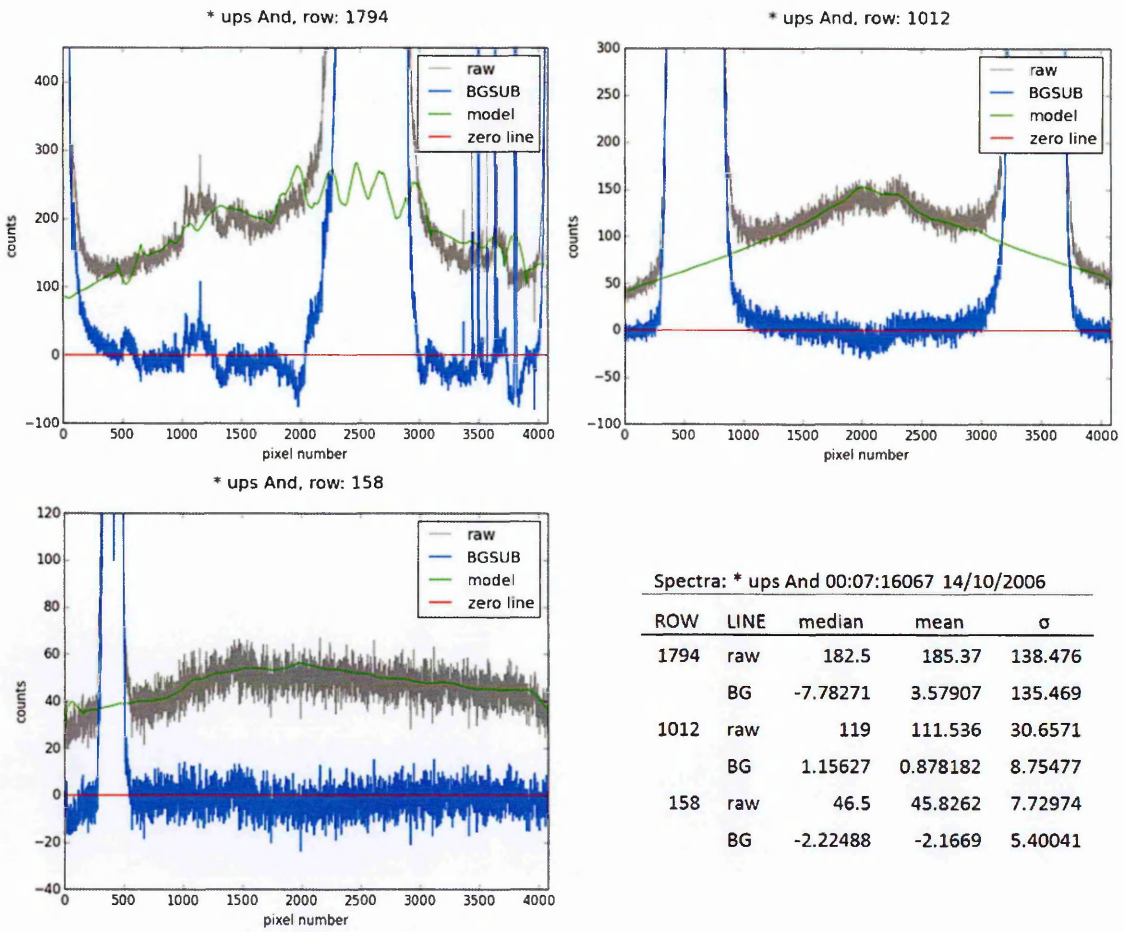


*Figure 27: CCD images of the Échelle spectrum of the star *ups And showing the un-subtracted raw exposure (left) and the background subtracted spectrum (right).*

As seen in the previous example, the Figure 27 before image (left) has a distinct shape to the background scattered light, emanating from the top centre of the chip and the after image (right) has a much more even background. However it is also noticeable that there

are image artefacts in the top of both images. These are much more irregular features than the main shape of the background and so have not been modelled and removed as effectively.



Spectra: * ups And 00:07:16067 14/10/2006

ROW	LINE	median	mean	σ
1794	raw	182.5	185.37	138.476
	BG	-7.78271	3.57907	135.469
1012	raw	119	111.536	30.6571
	BG	1.15627	0.878182	8.75477
158	raw	46.5	45.8262	7.72974
	BG	-2.22488	-2.1669	5.40041

*Figure 28: plots and data table, for the Star: * ups And, Date: 00:07:16067 14/10/2006. Top left: row: 1974, Top Right: row: 1012, Bottom left: row: 158, Bottom right: mean, median & std. deviation data table.*

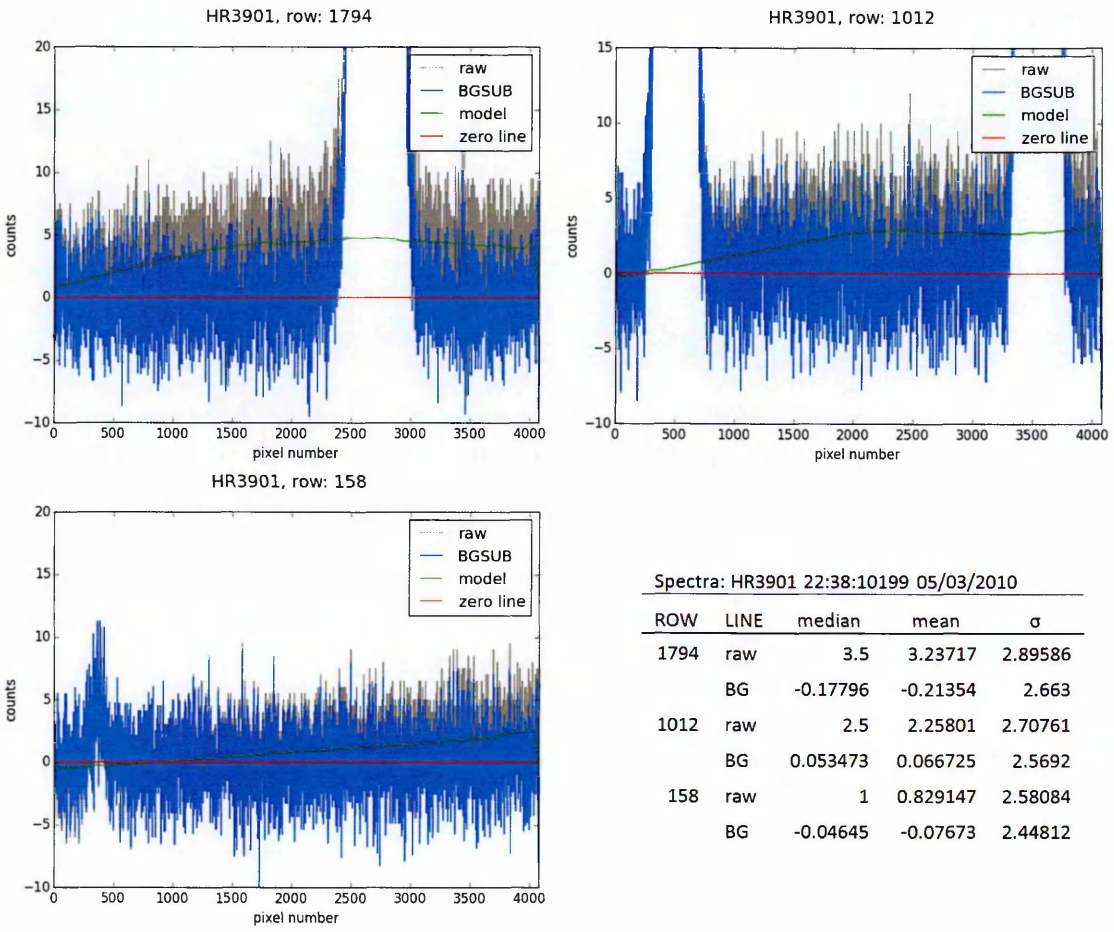
As with the previous high SNR spectrum we can see a reasonably good model fit and resulting subtraction for the lower rows, but in this case a poor fit for row 1794. This is due to the combination of the small mask order separation and the presence of the image artefacts visible in Figure 28. These artefacts contribute a more complex shape to the background and thereby exacerbating the poor fit. Specifically when the unmasked

background is elevated by any of these features, it will have the effect of pulling the smoothed model of the background away from the 'true' background. These features being abrupt deviations in the background are not well replicated in the smoothed model of the background and therefore not subtracted. The issue of a poor fit is made clear in plot of row 1794 (Figure 28 top left). Here the model line (green) is erratic, deviating above and below the raw (grey) line profile, resulting in over and under subtraction when the model is subtracted. Again the plots of rows 1012 and 158 show a good removal of the underlying background shape and the data table (bottom right) shows median and mean values close to zero counts, with low standard deviation showing the data to be flat.

Star: HR3901 SNR: 92 (LOW) observation date: 22:38:10199 05/03/2010

Figure 29: CCD images of the Échelle spectrum of the star HR3901 showing the un-subtracted raw exposure (left) and the background subtracted spectrum (right).

The before image in Figure 29 shows a much lower background noise level than the previous two high SNR examples. Again the after (right) image shows a much more even background suggesting that the shape has been effectively removed.



Spectra: HR3901 22:38:10199 05/03/2010

ROW	LINE	median	mean	σ
1794	raw	3.5	3.23717	2.89586
	BG	-0.17796	-0.21354	2.663
1012	raw	2.5	2.25801	2.70761
	BG	0.053473	0.066725	2.5692
158	raw	1	0.829147	2.58084
	BG	-0.04645	-0.07673	2.44812

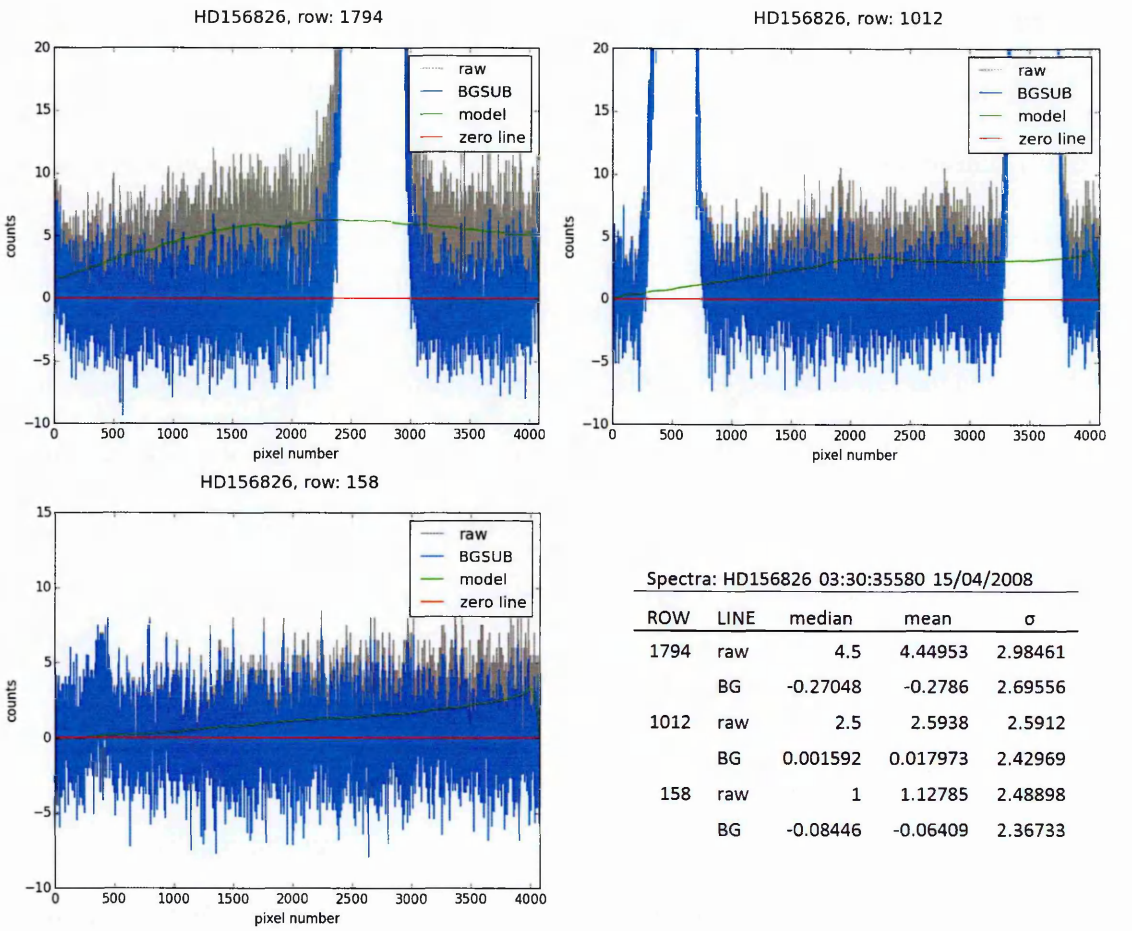
Figure 30: plots and data table, for the Star: HR3901, Date: 22:38:10199 05/03/2010. Top left: row: 1974, Top Right: row: 1012, Bottom left: row: 158, Bottom right: mean, median & std. deviation data table.

From Figure 30 we can immediately see that the level of counts in the raw data is much lower than those of the two high SNR spectra above. The model fits and subsequent reductions have resulted in much closer approximations to 0 counts for each row as shown in the plots and by the information given in the data table (Figure 30 bottom right).

Star: HD156826 SNR: 98 (LOW) observation date: 03:30:35580 15/04/2008

Figure 31: CCD images of the Échelle spectrum of the star HD156826 showing the un-subtracted raw exposure (left) and the background subtracted spectrum (right).

In Figure 31 a subtle low level noise background is visible in the before (left) frame, and the subtracted image (right) background appears more even.



Spectra: HD156826 03:30:35580 15/04/2008

ROW	LINE	median	mean	σ
1794	raw	4.5	4.44953	2.98461
	BG	-0.27048	-0.2786	2.69556
1012	raw	2.5	2.5938	2.5912
	BG	0.001592	0.017973	2.42969
158	raw	1	1.12785	2.48898
	BG	-0.08446	-0.06409	2.36733

Figure 32: plots and data table, for the Star: HD156826, Date: 03:30:35580 15/04/2008. Top left: row: 1974, Top Right: row: 1012, Bottom left: row: 158, Bottom right: mean, median & std. deviation data table.

Figure 32 shows that the background subtraction has worked well for this low SNR spectrum in each row. The Figure 32 data table shows the median and mean values for the subtracted backgrounds at approximately 0 counts, and also shows low standard deviation values indicating that the data is flat.

Section Conclusions

In the figures above it was seen that the background subtraction at row 1794 was not always accurate. This was due to the masked Échelle orders in this wavelength region, corresponding to the red end of the visible spectrum, being much closer in and therefore fewer inter-order left unmasked for the background model to fit. This however does not happen in the lower orders, corresponding to the blue end of the spectrum. Here the order spacing is much greater so there is much more background to be modelled, resulting in a much better background fit and consequent subtraction. The more accurate the model of the background, the more accurate the subtraction. This was seen for all spectra tested at row = 158, as demonstrated by the plots and accompanying data tables (Figures 26, 28, 30 and 32).

Fortunately for these investigations the Ca II H and K lines are located on two of the lower most orders on the chip, namely the second and third orders. We would therefore expect the background subtraction to be reliable for these orders. To check this, the region of the chip containing the Ca II H and K line cores was investigated. Figure 33 below, shows 3-D surface plots for the image data of the *eta Cas spectra used above. This serves as a visual demonstration of the background subtraction in this region. In the top panel it can be seen that the initial raw data has a non-zero background contribution, where the red plane represents the zero level. The background level has then been removed in the bottom panel as we can see the blue surface plot covering the red zero-level plane.

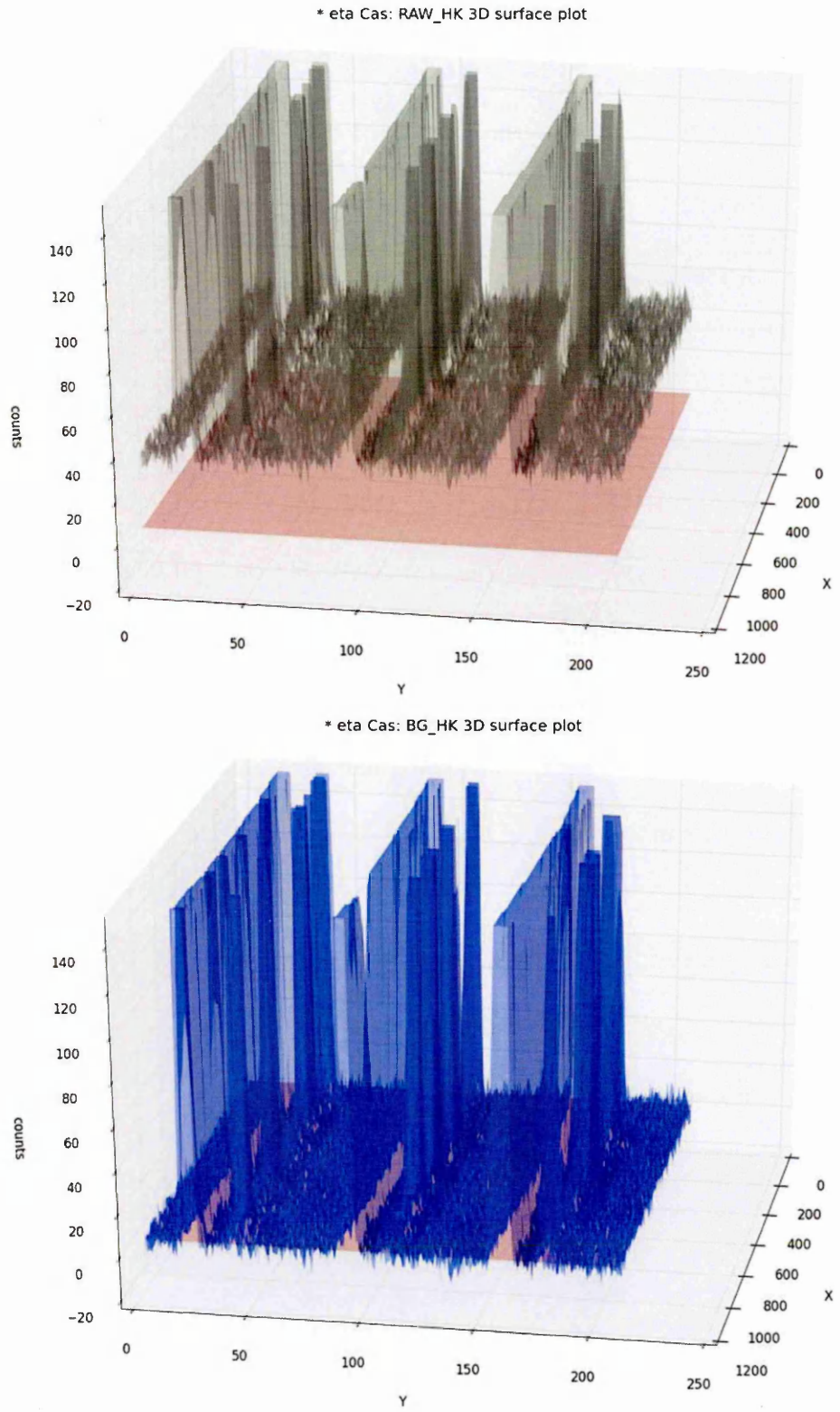


Figure 33: zero level is represented by the red plane, the raw un-subtracted image data is shown as the grey surface plot (top panel) and the background subtracted image data is shown by the blue surface plot (bottom panel)

STAR	IMAGE	median	mean	σ
*eta Cas	RAW_HK	39	39.1185	5.22531
	BG_HK	0.184458	0.309517	4.61821
* ups And	RAW_HK	48.5	49.7807	13.1747
	BG_HK	-0.76514	0.564546	12.9961
HR3901	RAW_HK	1.5	1.35414	2.45379
	BG_HK	-0.02192	-0.0078	2.44504
HD156826	RAW_HK	1.5	1.6387	2.37572
	BG_HK	0.039928	0.031647	2.3637

Table 4: Ca II H and K regions: $x_{min} = 2300$, $x_{max} = 3400$, $y_{min} = 40$, $y_{max} = 250$

Table 4 shows the resulting median, mean and standard deviation values of the background in this region for both the raw (RAW_HK) and subtracted (BG_HK) images. For all stars it can be seen that the median background level has been reduced to effectively zero in the subtracted images. Additionally, the low standard deviation values in this region attest to the background being flat. This validates the effectiveness of the SKYBM software in this region.

13.2 ECHOMOP COMPARISON TESTS

In the previous section we established that the SKYBM software does provide an effective background subtraction, being most accurate in the lower regions of the CCD chip where the Échelle orders are further spaced. Conveniently this also happens to be the region in which the Ca II H and K lines, of interest to our investigations, are located. We then sought to compare the effectiveness of our software's background subtraction to those offered by the established ECHOMOP software, namely Options 6 and 22 (see Section 11).

Since the Ca II H and K lines are confined to the lower region of the CCD chip and because the SKYBM subtraction is most accurate in this region, only the area 300 pixels from the base of the chip will be used for each spectrum. Trimming the Échelle spectra frames to this extent also carried the advantage of reducing the number of orders being processed from 82 (41 spectral orders and 41 calibration orders, see Figure 15) to just 8 (4 spectral orders and 4 calibration orders), greatly reducing the computational time. The trimming of the CCD image frames had to be down prior to operation for use in ECHOMOP, however for SKYBM this could be done internally, see Section 12.3.

It is not possible to inspect the flux levels in the regions between or beneath orders in ECHOMOP in the way that were able to for SKYBM. This means that there is no opportunity to inspect the background noise level after the Échelle reduction process. Instead we could only assess the quality of the background subtraction by inspecting and comparing the resultant reduced spectra at the end of the ECHOMOP pipeline. The ECHOMOP Échelle data reduction procedure used in this work was described in Section 11 which has the end result of producing collapsed Échelle spectra such as that shown in Figure 15.

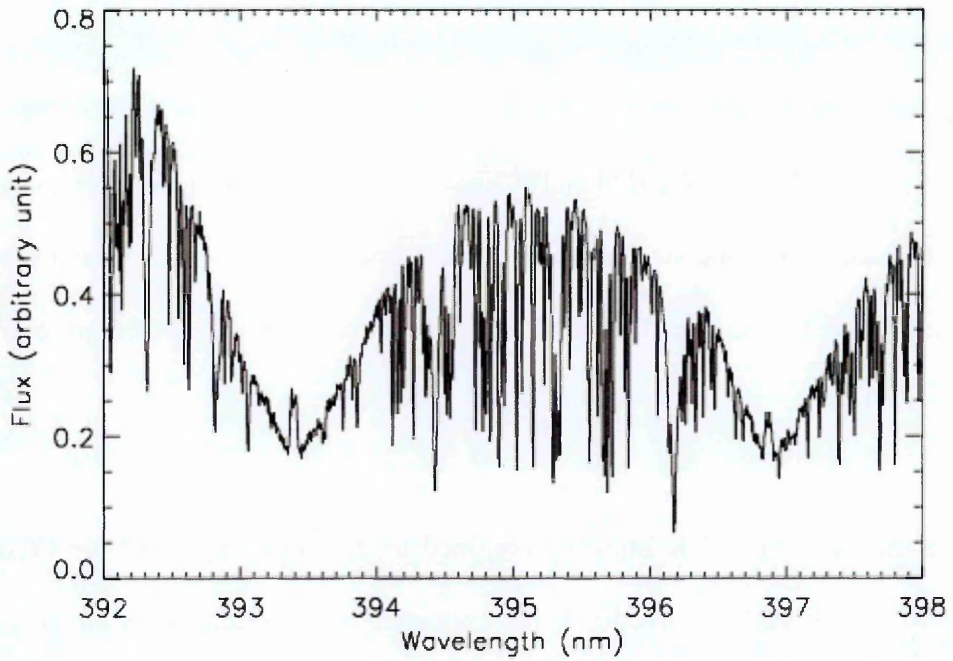


Figure 34: The stellar spectrum of the star HD 9446. This region shows the Ca II H [3968.5 Å] and K [3933.7 Å] resonance lines, where the line cores and the corresponding central emission peaks are plainly visible. Image source: (Hebrard et al. 2010)

Once a collapsed spectrum had been produced it was a simple matter of extracting specific horizontal rows (orders) to obtain the spectral data of interest. The spectral region of interest to our investigations is that of the Ca II H and K resonance lines, shown in Figure 34. In the SOPHIE Échelle spectra these line core regions are found on two separate orders near the bottom of the chip.

It is difficult to judge the accuracy of the background subtraction from these collapsed Échelle spectra as they do not contain any of the inter order data previously used to assess the background level. It is therefore not possible to assess whether the spectra produced using different methods have had their backgrounds over, under or adequately subtracted.

One way to compare the accuracy of different subtraction methods would be to use an Échelle spectrum which has already had its background satisfactorily removed. Applying the ECHOMOP procedure would then give the collapsed Échelle spectrum as would be the result had the spectrum had no initial background. This would then serve as a reference subtraction target allowing us to judge how accurately the background had been removed by the other options.

In the previous Section the backgrounds of four test spectra (of the stars * eta Cas, * ups And, HR3901, HD156826) were satisfactorily removed in the region containing the Ca II H and K lines. This was supported by the numerical values in Table 4. The trimmed Gaussian background, shown in Figure 35, was then introduced to all four trimmed spectra frames previously flattened by SKYBM. This background shape was deliberately exaggerated to make the differences between the subtraction methods outputs more pronounced. These 'background added' spectra were then fully reduced using ECHOMOP to produce four collapsed Échelle spectra for each of the subtraction options SKYBM, Opt 6 and Opt 22 including 'No subtraction' (where the artificial background spectrum was used but with no subsequent background subtraction method applied to it). ECHOMOP was used again to produce the collapsed Échelle spectrum for the 'flat' spectrum (where 'flat' indicates the previously subtracted spectrum before the artificial background was introduced). In the case of the SKYBM, as described in Section 12.3, the subtraction was conducted prior to use in ECHOMOP.

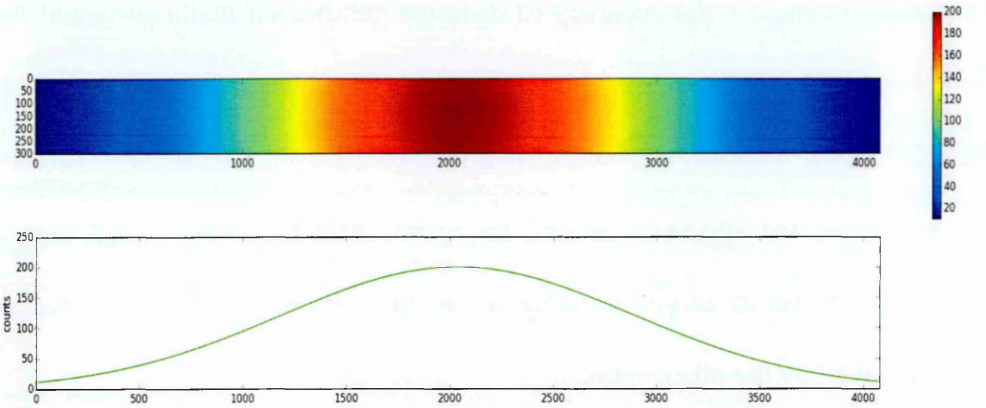
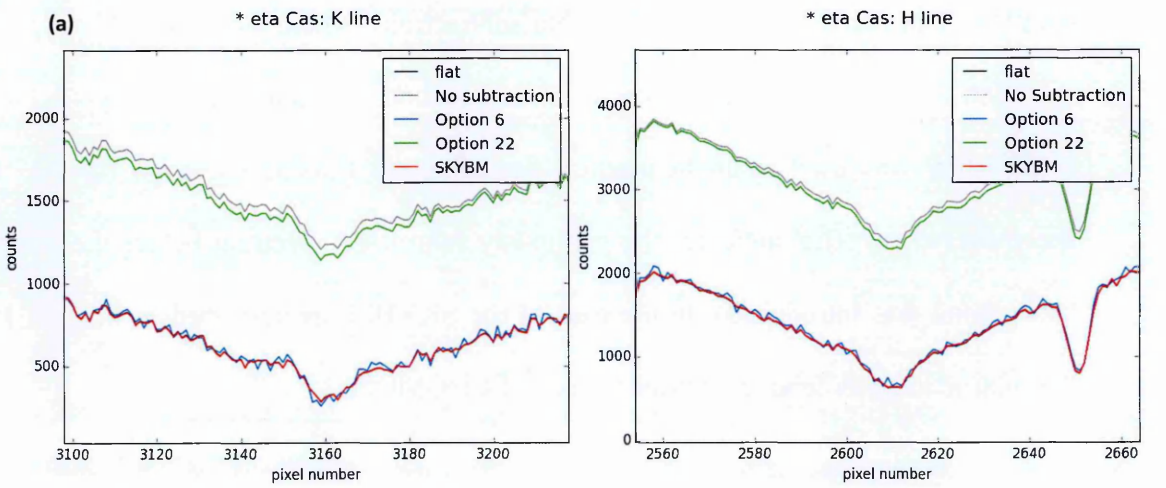


Figure 35: Trimmed Gaussian background introduced to the four trimmed 'flattened' spectra to compare background subtraction methods. Top panel shows the 2D surface profile of the background. Bottom panel shows the cross-section profile at row 150.

Proof of an accurate background subtraction would be to return the flux level of the spectral orders to those of the 'flat' starting point spectrum. Using a short Python script the specific orders containing the Ca II lines were extracted for each stellar spectrum and background subtraction method, the comparative spectra were then displayed using Matplotlib in the plots shown in Figure 36 below.



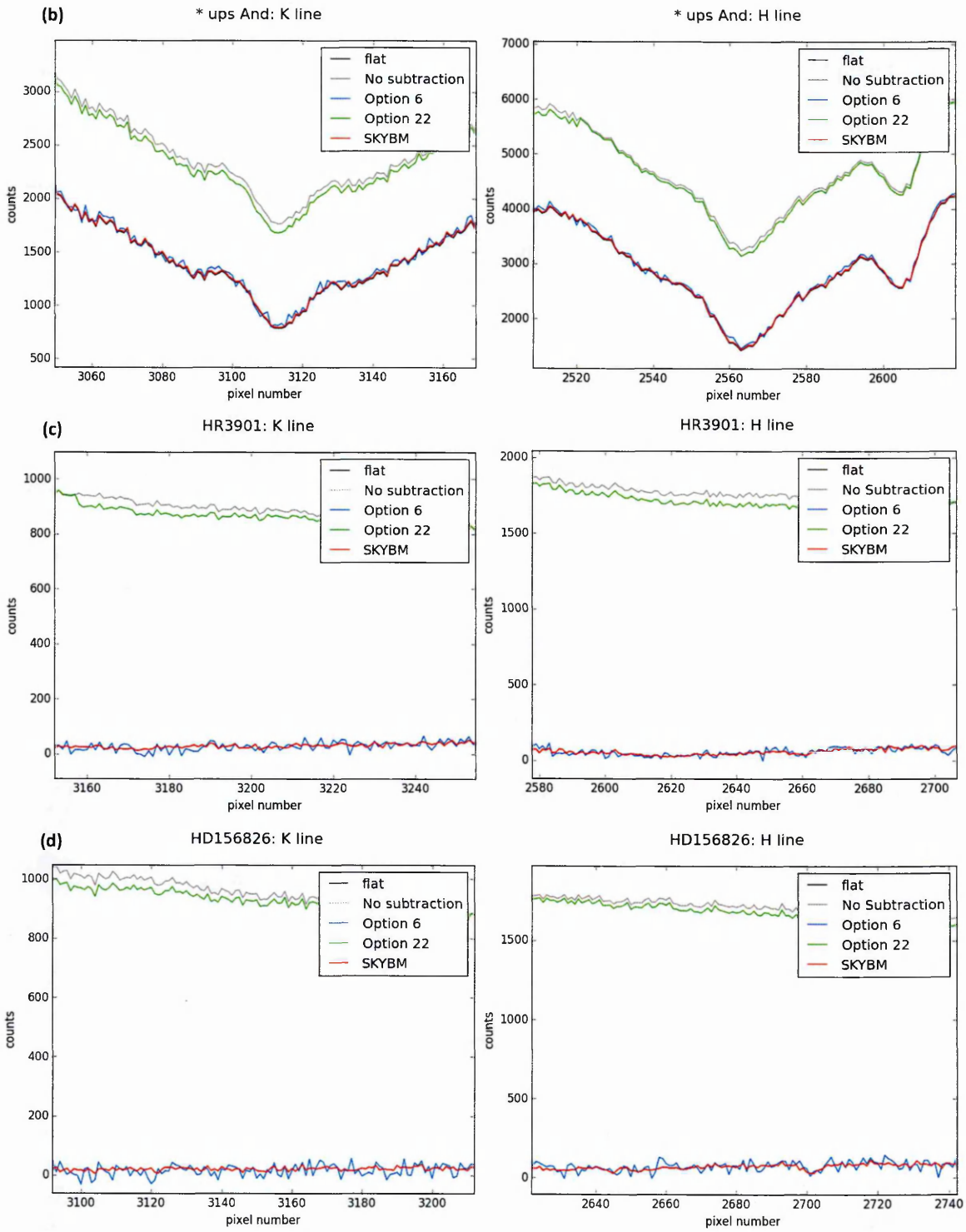


*Figure 36: (a-d) Ca II K (left panels) and H (right panels) spectral regions for each of the four stellar spectra from the stars *eta Cas, *ups And, HR3901 and HD156826. Shown in each plot are the reduced spectra produced using the background subtraction methods: ECHOMOP Option 6 (blue), ECHOMOP Option 22 (green) and the SKYBM software (red). Also shown are the spectra produced when no subtraction method was used (grey) as well as the 'flat' previously subtracted Échelle spectra. The axes have been scaled to show the line profiles of each subtraction method on the same plot.*

The plots in Figure 36 above show the reduced spectra for each of the different subtraction options discussed previously, in the regions of the Ca II H and K lines. Each plot displays the spectra resultant from the three different background subtraction methods used, ECHOMOP's Option 6 in blue, ECHMOP's Option 22 in green and SKYBM in red.

The grey line shows the reduced spectra of raw data, where no background subtraction has been applied. This line serves as a point of reference, visually demonstrating the effect each subtraction method has had on the data. Also shown are the reduced spectra of the 'flat' (previously subtracted) Échelle frames, in black. The black 'flat' line provides a reference reduction target showing the true level of the spectra before the artificial background was added. The goal for an accurate subtraction would be therefore to reduce the background elevated spectra to this level.

The black line is difficult to see on the scale shown in these plots, due to its proximity to the red SKYBM and blue Opt 6 lines. Figure 37 provides a closer look at this region for the K line plot of the star * ups And. This is the case for all plots shown.

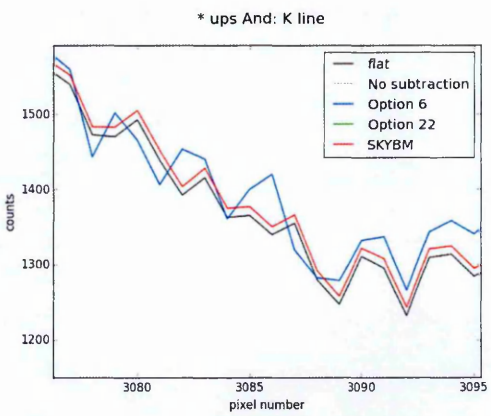


*Figure 37: smaller region of the * ups And K line plot, providing a closer view of the Option 6, SKYBM and flat lines. On this scale the separation between the lines is made more apparent.*

Section Conclusions

The plots in Figure 36 show consistent outcomes for each of the background subtraction options used. In all cases, Option 6 had the result of decreasing the overall level of the spectra in the regions shown. Comparing the blue lines to those of the grey and black lines shows that Option 6 consistently produces spectra matching the overall level of the target 'flat' lines, indicating that it is accurately accounting for and subtracting the background noise. However it is also apparent that the Option 6 lines are noisier than either the grey or the black lines. Judging from the form of these reference lines this noise must have been introduced by this option and does not reflect the accurate profile of the spectrum. In comparison to the other subtraction methods, Option 22 has only brought about a small level of subtraction and is far from the target level given by the black line. In contrast to Option 6, Option 22 does not introduce any additional noise to the spectral line profile but maintains the form.

As detailed in Section 11, the two ECHOMOP options take different approaches to modelling and removing the background, which is the likely cause of the differences we are seeing in their outputs. Option 6 makes local approximations of the background contribution along each order but does not reference the surrounding regions. This will more closely trace the overall level of the background at any given point but it will pick up more local variations making for a more erratic background approximation which is then reflected in the output. Option 22 by modelling all the regions between orders will produce a smoother global approximation of the background but can thereby smooth out important regional subtleties. This may not be an issue on the scale of the entire

spectrum, but given the relatively smaller wavelength regions that we are concerned with this will be important.

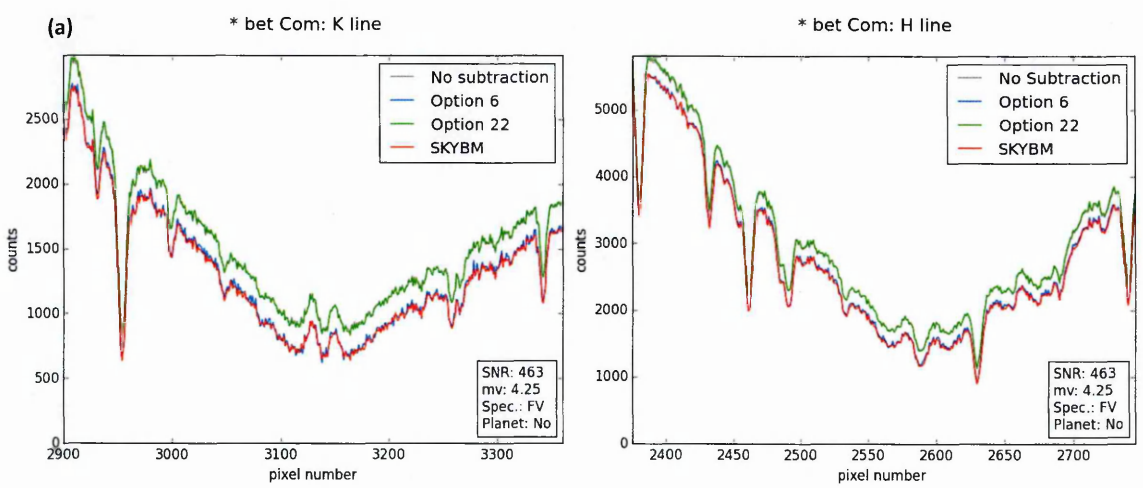
In all cases shown the red SKYBM lines are the closest match to the black target lines. Unlike Option 6, SKYBM does not introduce any additional noise maintaining the form of the spectra we see in the two un-subtracted reference lines. This is because SKYBM uses a polynomial filter (see Section 12.3) to make a smooth approximation of the background and will thus not be as affected by local variations. In comparison with Option 22, SKYBM brings about an accurate level of subtraction consistently closely matching, and in some cases overlapping the black target line. Unlike Option 22, SKYBM models the background beneath orders and so will more accurately capture the background level in the region of the spectral data. These findings demonstrate that of the three background subtraction options used, SKYBM performs the most accurate subtraction of the background in this region. This suggests that this option will be adequate to subtract the background for subsequent log R'_{HK} calibration.

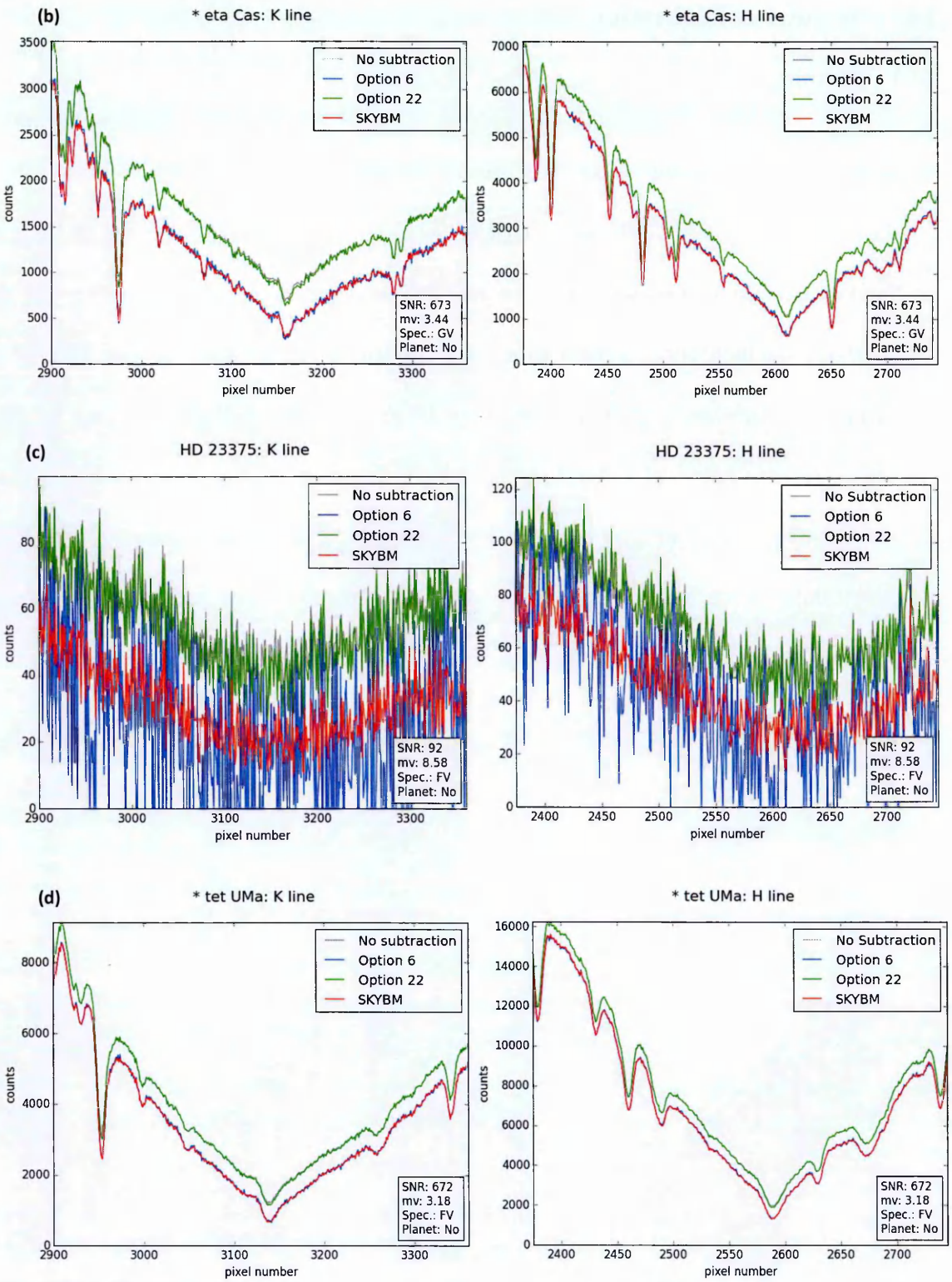
14. Results & Discussion

14.1 Results

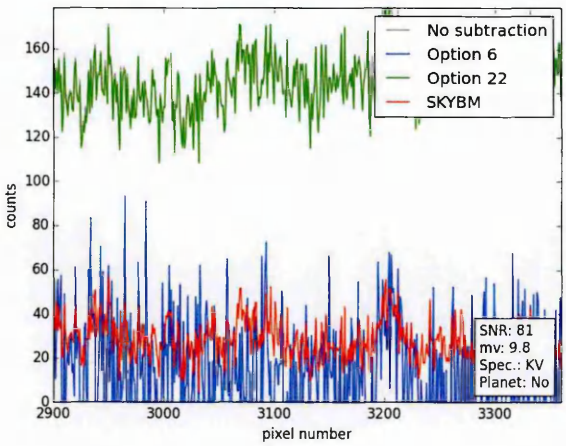
In the previous Section we performed detailed comparisons of the background subtractions produced using the three different background subtraction methods ECHOMOP Option 6:ech_sky, ECHOMOP Option 22:ech_mdlbck and *SKYBM* (our background subtraction software,) for just four test Échelle spectra. To further test these methods we then applied them to a larger sample of stellar spectra. The Échelle data reduction procedure (detailed in Sections 10 & 11) was conducted for each of the 24 stellar spectra listed in Table 3 using three different background subtraction methods resulting in 72 background subtracted collapsed Échelle spectra. Using a short Python script these orders were extracted for each stellar spectrum and background subtraction method, and the comparative spectra displayed using Matplotlib in the plots shown in Figure 38 below.

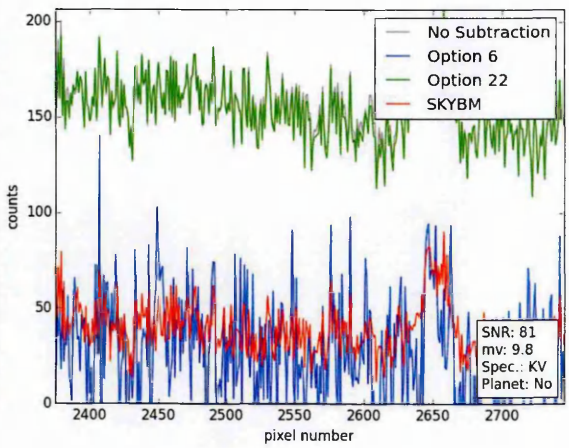
Non planet-hosts:



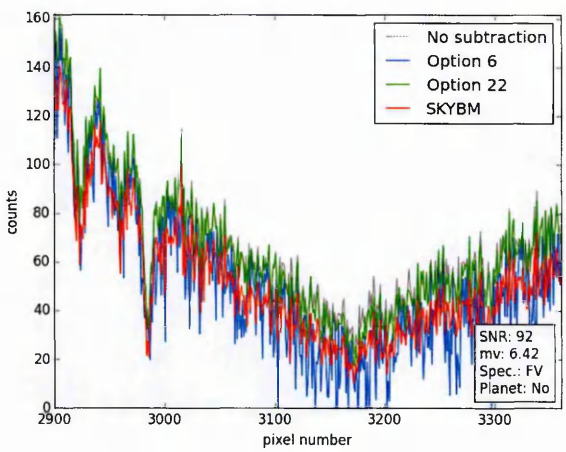

(b)
* eta Cas: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 673
mv: 3.44
Spec.: GV
Planet: No
counts
pixel number
* eta Cas: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 673
mv: 3.44
Spec.: GV
Planet: No
counts
pixel number
(c)
HD 23375: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 92
mv: 8.58
Spec.: FV
Planet: No
counts
pixel number
HD 23375: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 92
mv: 8.58
Spec.: FV
Planet: No
counts
pixel number
(d)
* tet UMa: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 672
mv: 3.18
Spec.: FV
Planet: No
counts
pixel number
* tet UMa: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 672
mv: 3.18
Spec.: FV
Planet: No
counts
pixel number

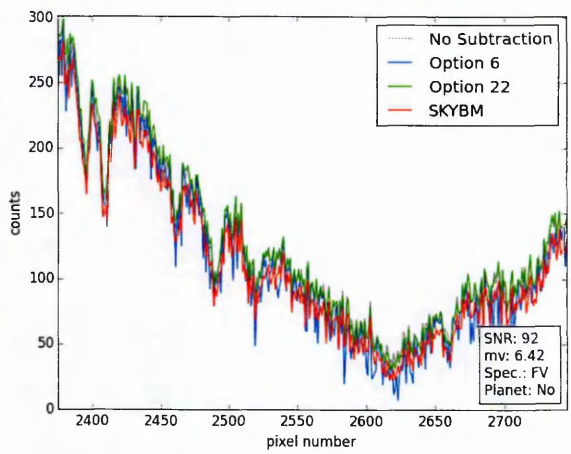
(e)


HD 96692: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 81
mv: 9.8
Spec.: KV
Planet: No
counts
pixel number

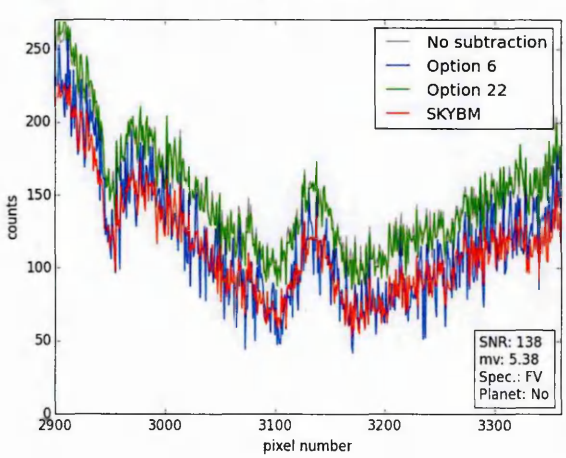

HD 96692: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 81
mv: 9.8
Spec.: KV
Planet: No
counts
pixel number

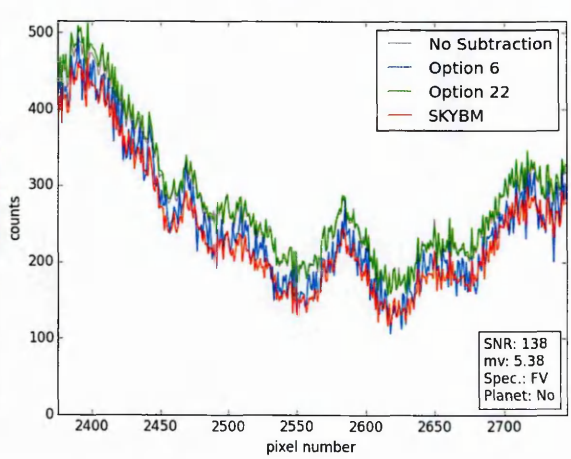
(f)

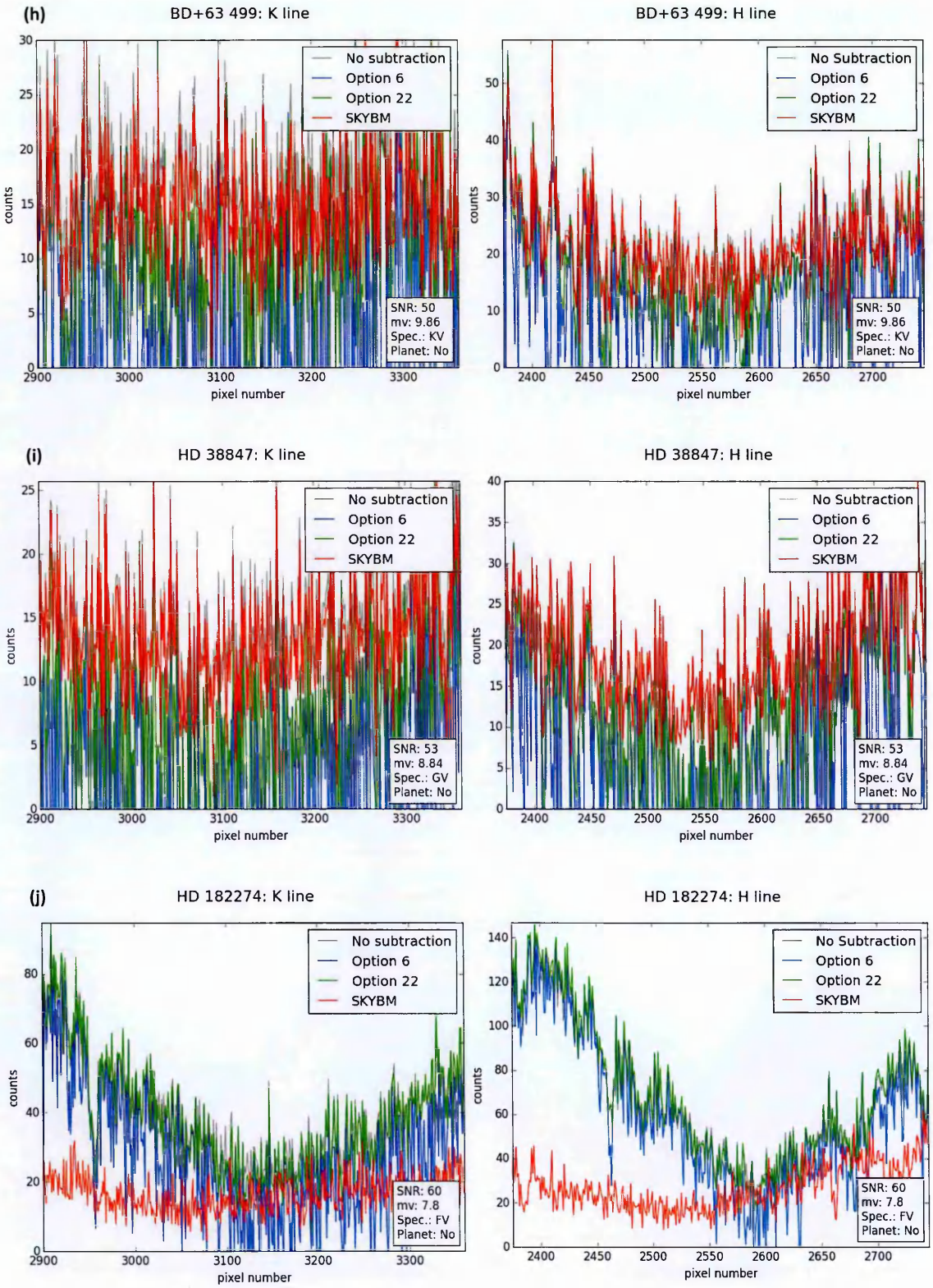

HR 3901: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 92
mv: 6.42
Spec.: FV
Planet: No
counts
pixel number

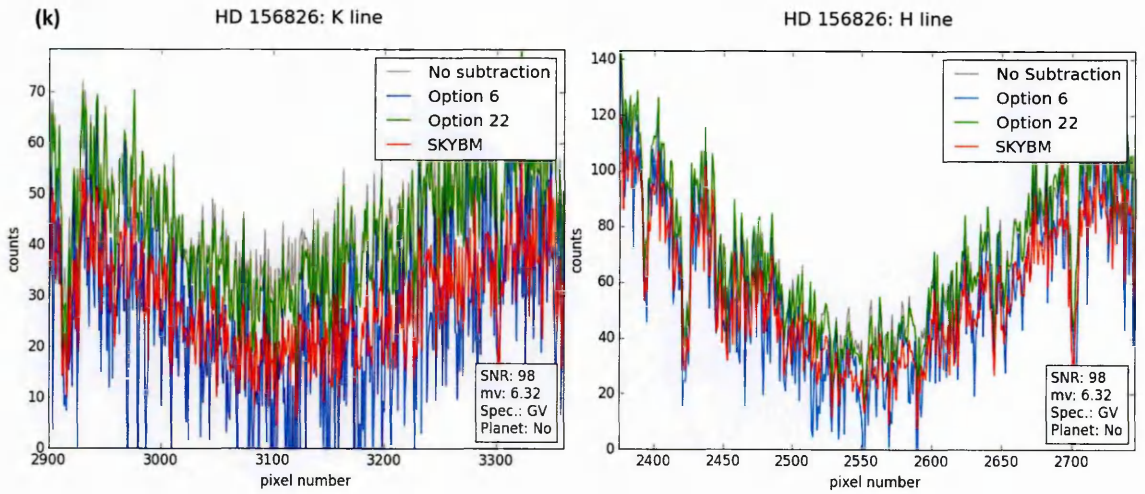

HR 3901: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 92
mv: 6.42
Spec.: FV
Planet: No
counts
pixel number

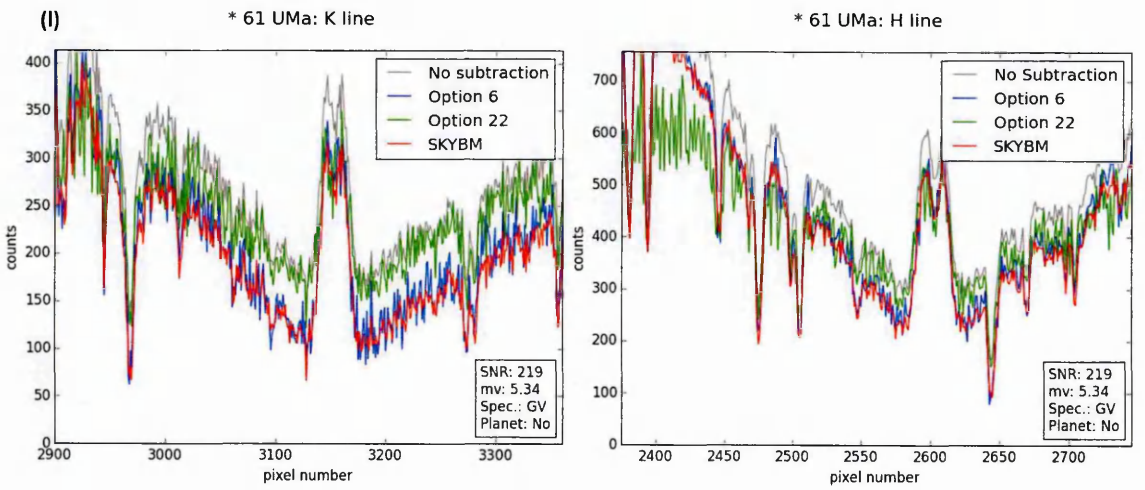
(g)


HD 25457: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 138
mv: 5.38
Spec.: FV
Planet: No
counts
pixel number

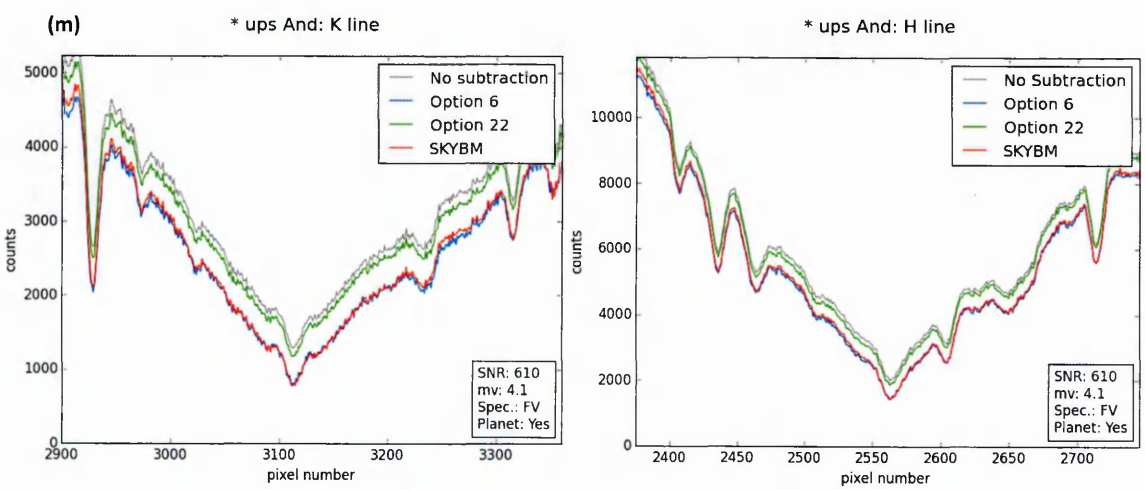

HD 25457: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 138
mv: 5.38
Spec.: FV
Planet: No
counts
pixel number

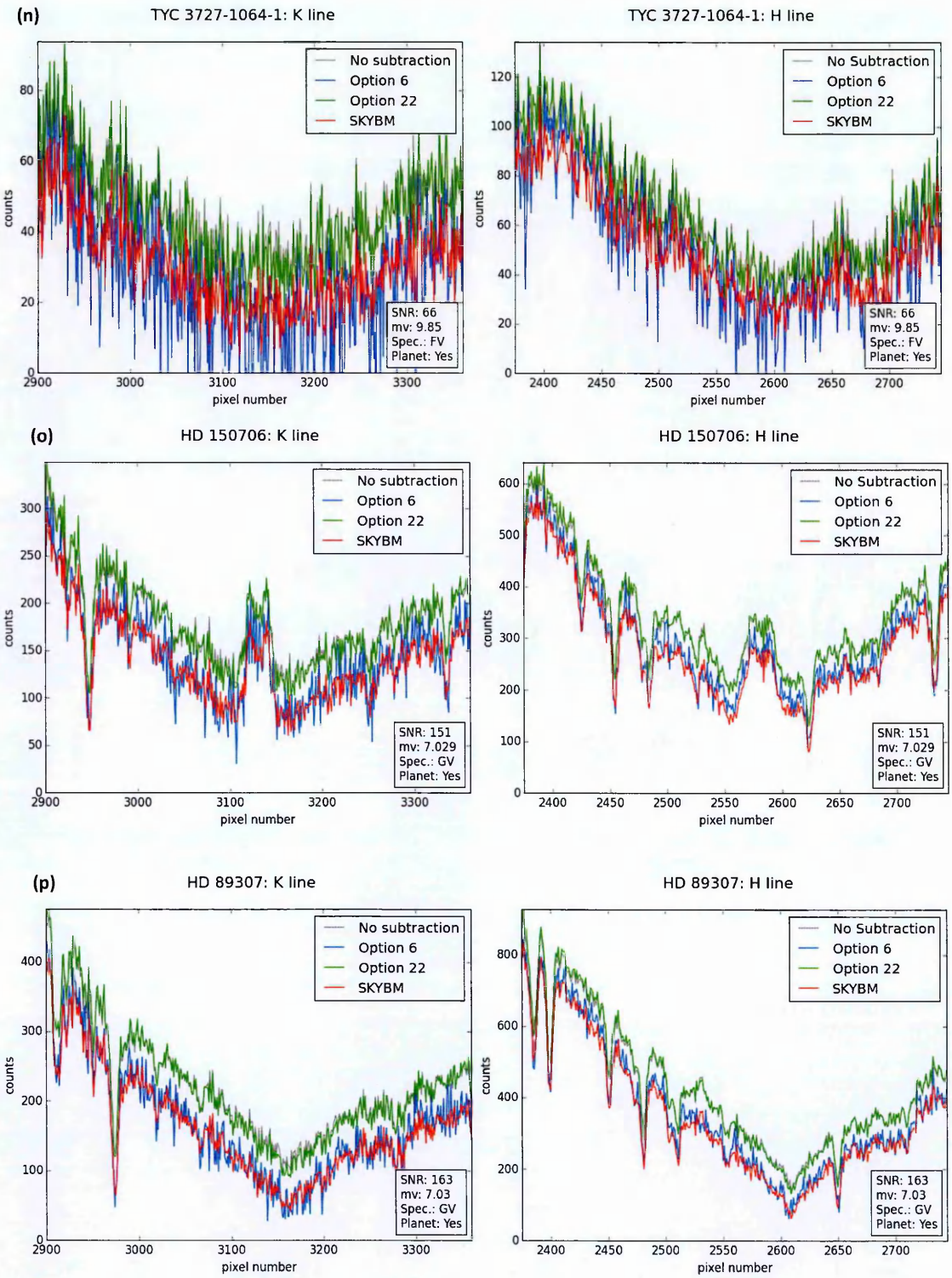

(h)
BD+63 499: K line
BD+63 499: H line
No subtraction
Option 6
Option 22
SKYBM
SNR: 50
mv: 9.86
Spec.: KV
Planet: No
counts
pixel number
(i)
HD 38847: K line
HD 38847: H line
SNR: 53
mv: 8.84
Spec.: GV
Planet: No
(j)
HD 182274: K line
HD 182274: H line
SNR: 60
mv: 7.8
Spec.: FV
Planet: No



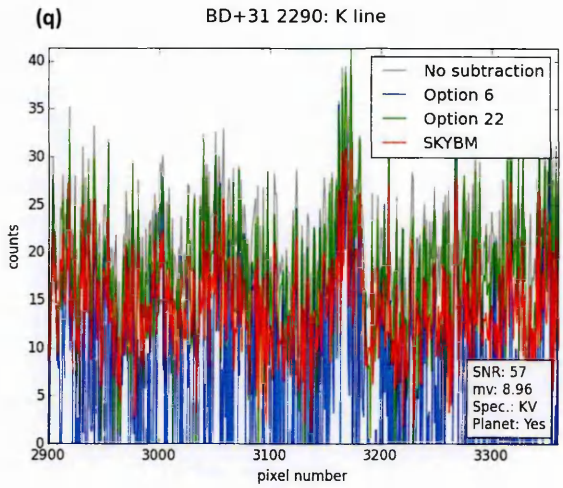


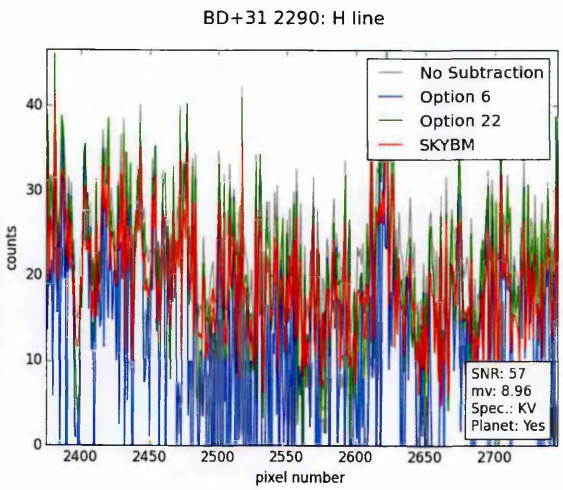
Planet-hosts:



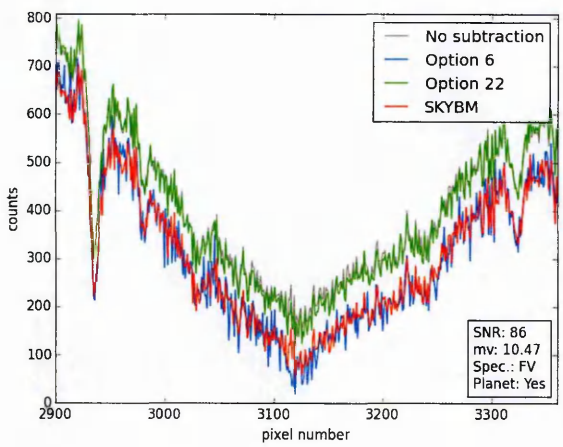

(n)
TYC 3727-1064-1: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 66
mv: 9.85
Spec.: FV
Planet: Yes
counts
pixel number
TYC 3727-1064-1: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 66
mv: 9.85
Spec.: FV
Planet: Yes
counts
pixel number
(o)
HD 150706: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 151
mv: 7.029
Spec.: GV
Planet: Yes
counts
pixel number
HD 150706: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 151
mv: 7.029
Spec.: GV
Planet: Yes
counts
pixel number
(p)
HD 89307: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 163
mv: 7.03
Spec.: GV
Planet: Yes
counts
pixel number
HD 89307: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 163
mv: 7.03
Spec.: GV
Planet: Yes
counts
pixel number

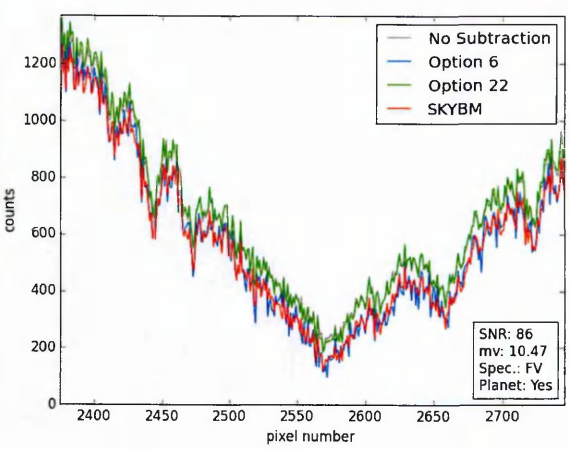
(q)


BD+31 2290: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.96
Spec.: KV
Planet: Yes
counts
pixel number

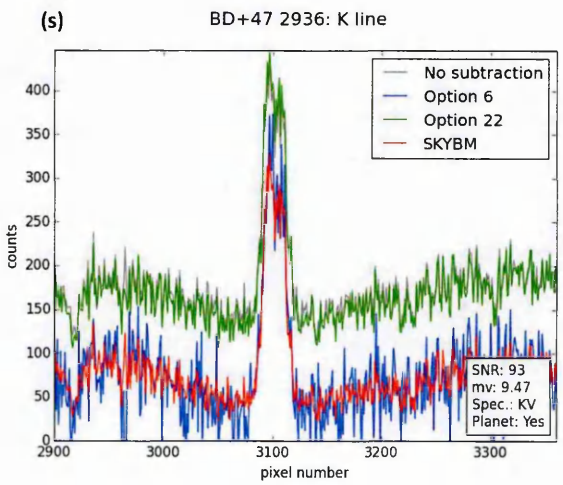

BD+31 2290: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.96
Spec.: KV
Planet: Yes
counts
pixel number

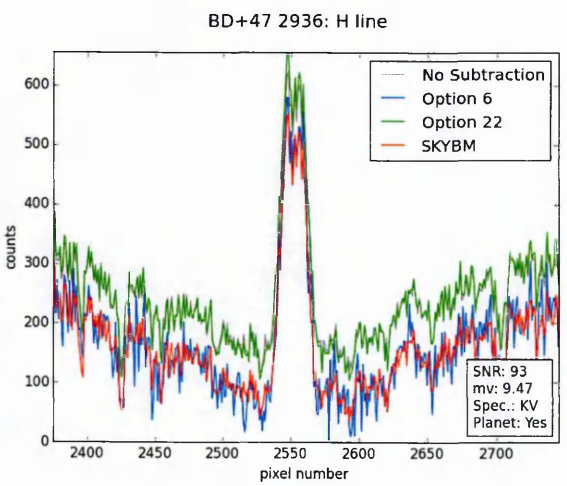
(r)

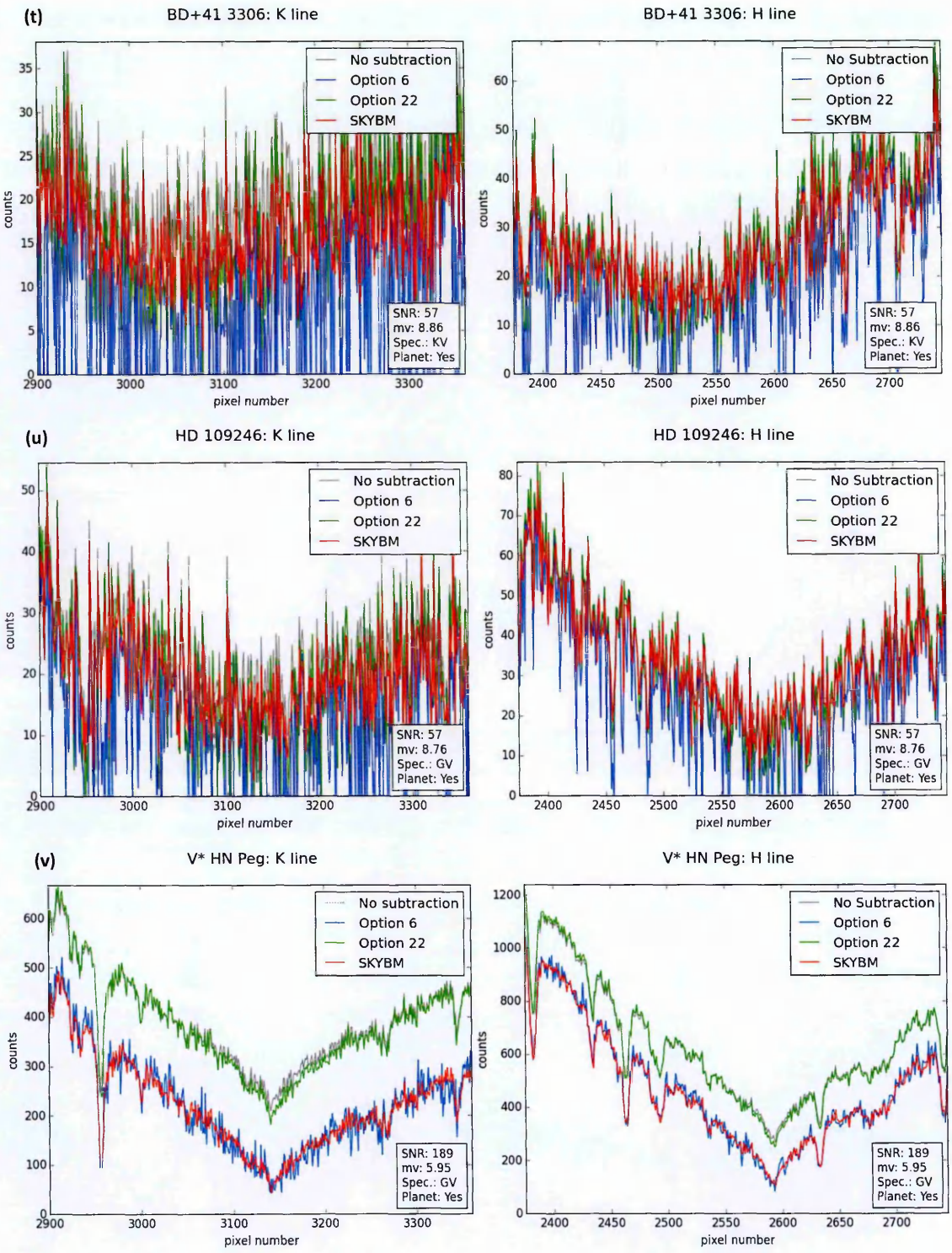

TYC 3239-992-1: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 86
mv: 10.47
Spec.: FV
Planet: Yes
counts
pixel number


TYC 3239-992-1: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 86
mv: 10.47
Spec.: FV
Planet: Yes
counts
pixel number

(s)


BD+47 2936: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 93
mv: 9.47
Spec.: KV
Planet: Yes
counts
pixel number


BD+47 2936: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 93
mv: 9.47
Spec.: KV
Planet: Yes
counts
pixel number


(t)
BD+41 3306: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.86
Spec.: KV
Planet: Yes
counts
pixel number
BD+41 3306: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.86
Spec.: KV
Planet: Yes
counts
pixel number
(u)
HD 109246: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.76
Spec.: GV
Planet: Yes
counts
pixel number
HD 109246: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 57
mv: 8.76
Spec.: GV
Planet: Yes
counts
pixel number
(v)
V* HN Peg: K line
No subtraction
Option 6
Option 22
SKYBM
SNR: 189
mv: 5.95
Spec.: GV
Planet: Yes
counts
pixel number
V* HN Peg: H line
No Subtraction
Option 6
Option 22
SKYBM
SNR: 189
mv: 5.95
Spec.: GV
Planet: Yes
counts
pixel number

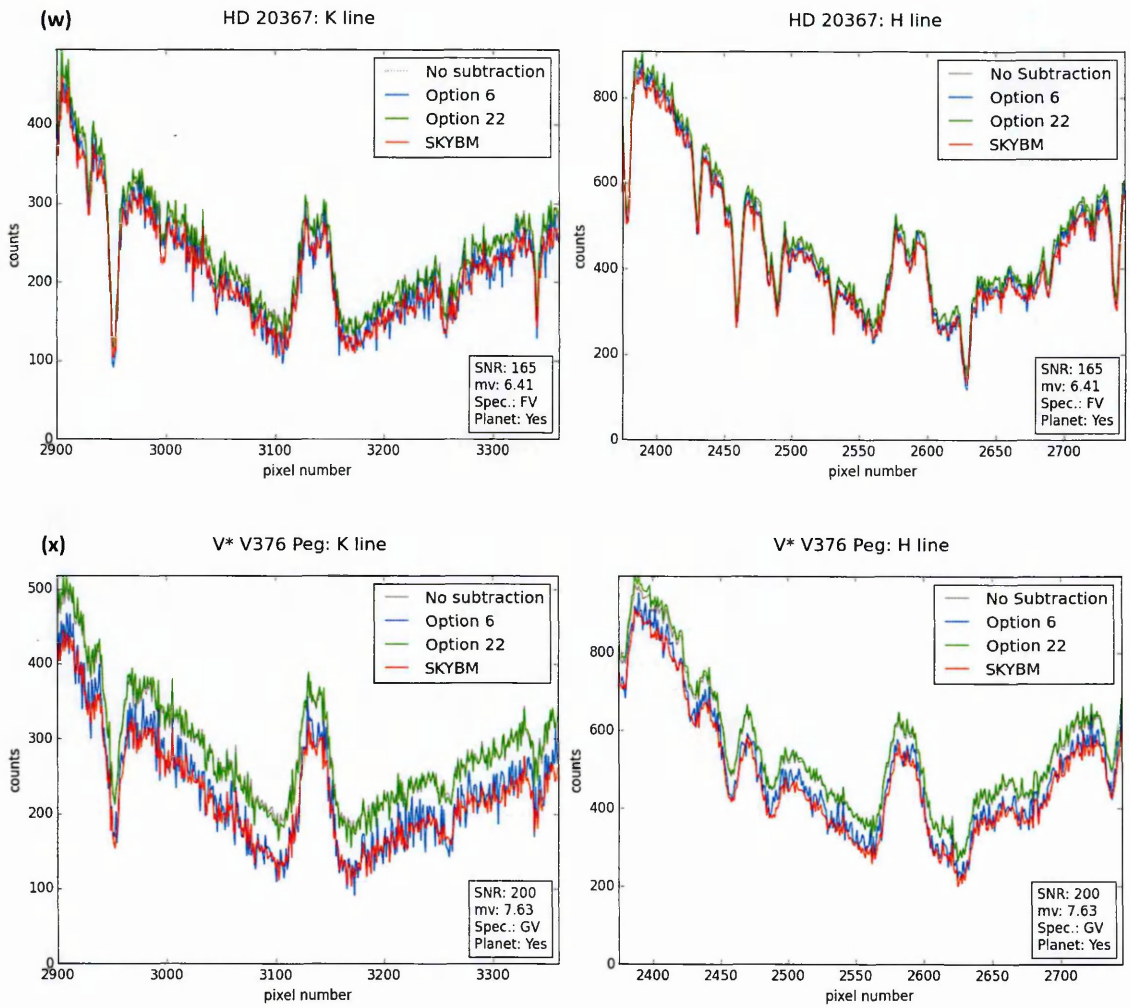


Figure 38: (a-x) Ca II H (left panels) and K (right panels) spectral regions for each of the spectra listed in Table 3. Shown in each plot are the reduced spectra generated using the ECHOMOP Option 6 (blue), ECHOMOP Option 22 (green) and the SKYBM software (red). Also shown are the raw un-subtracted spectra for comparison (grey). Additionally information of the signal-to-noise ratio (SNR), visual apparent magnitude (mv), spectral type (Spec.) and known planet hosting status are shown in the bottom right corners of each plot.

The plots in Figure 38 above show the reduced Échelle spectra in the regions of the Ca II H and K resonance lines for the selection of stellar spectra listed in Table 3. Each plot displays the spectra resultant from the three different background subtraction methods used, ECHOMOP's Option 6 in blue, ECHMOP's Option 22 in green and SKYBM in red. Also shown are the reduced raw spectra in grey, where no background subtraction has been applied, in similar fashion to Figure 36 of the previous Section. These plots

show a variety of outcomes in terms of quality of spectra and extent of reduction. Ideally the background subtraction should have the effect of subtracting the underlying light contribution, without losing any spectral information.

14.2 Discussion

Signal-to-noise (SNR) & visual apparent magnitude (mv)

It is immediately apparent that the signal-to-noise ratio (SNR) has had a significant impact on the outcome in terms of the quality of the spectra. In general, spectra with SNR below 100 (in particular Figure 38: e, h, i, q) are so noisy that it is difficult to distinguish between the different subtraction modes or to identify spectral features. This low SNR correlates with higher magnitude (i.e. dim stars) making it all the more important that higher SNR are achieved for these candidates.

In contrast spectra with SNR above this tentative basal limit are of sufficient quality that the underlying form of the spectrum can be observed with clear distinction between the different subtraction modes (Figure 38: b, d, m and a are notable examples). It is not possible from this sample size to provide an exact numerical lower limit of SNR, however it is advisable that the SNR of any future candidates for this reduction procedure exceed 100, with even higher being more desirable.

ECHOMOP background subtraction options

The two ECHOMOP background subtraction options demonstrated noticeably different yet consistent outcomes in their resulting spectra. These were the same outcomes observed, detailed and explained in the previous Section. Specifically, Option 6 produces

noisier spectra yet a greater level of subtraction, relative to the grey line, whereas Option 22 produces a much smaller subtraction, but does not add any noise effects to the output. In the previous Section we were able to judge the accuracy of the level of subtraction using a reference spectrum where the background had already been carefully subtracted. This would not be practical to do for large numbers of stellar spectra as it would require knowing precisely background scattered light contribution beforehand which is the very problem that background subtraction software seeks to address. It was seen, for the four test spectra, that the Option 6 yielded an accurate level of reduction suggesting that the subtractions of the output spectra shown in Figure 38 are also accurate.

For many of the plots, in Figure 38, the Option 22 output spectra match the grey line so closely that no subtraction is apparent. This is consistent with the findings of Section 13.2 which also saw much smaller subtractions from this option. From those tests we saw that Option 22, rather than having no affect, did result in a level of subtraction in every case. The difference was much more noticeable however in Section 13.2 due to the exaggerated artificial background used for demonstration and comparison purposes. In those previous tests, Option 22 was found to be under-approximating the background contribution by not subtracting to the extent of the Option 6 or the SKYBM level (which were deemed more accurate). This is seen again in the plots of Figure 38. These two outcomes are most obvious in Figure 38.x where the Option 6 data (blue) is at a lower level but is noisier than Option 22 data (green).

SKYBM

In all but one of the plots in Figure 38, the SKYBM subtraction data (in red) matched the level of subtraction achieved by Option 6, whilst maintaining the noise level of the

original input spectra. Again this is consistent with the results of Section 13.2. Of the three methods tested in Section 13.2, it was found that SKYBM provided the most accurate background subtractions in this region. Because of this we would also expect the above subtractions to be accurate. This is supported by the output spectra of Option 6, which was shown to give an accurate level of background subtraction in our tests (Section 13.2). If we can trust the outputs of Option 6 (being part of established software) to be consistent in the above plots then the fact that SKYBM matches the level of subtraction throughout (in all usable reductions) is evidence of its accuracy.

We suspect that the accuracy of the SKYBM subtraction is due to it utilising more of the data provided from the CCD chip in generating the background model than the other two options as explained in Sections 11 and 13.2. By utilising more information a more accurate estimate of the local and global background contribution is achievable. This would ensure that local low level data are not artificially smoothed out and also that inherent noise fluctuations do not deform the model. SKYBM would seem therefore to present the best option for the background subtraction in all the spectra studied with the exceptions of the very low SNR plots, where no best option is apparent, and also notably in the case shown in Figure 38.j. Here the spectrum differs greatly from those of the other options suggesting that a problem has occurred with the SKYBM subtraction. Upon inspection it was found that this problem was the result of an unsuitable mask being created and subsequently used. Specifically this resulted in spectral order data being un-masked and therefore removed with the background subtraction. This emphasises the need to ensure that the masks being used are suitable.

15. Project Conclusions

The intention of this project was to construct a reliable software pipeline for reducing Échelle spectra from the SOPHIE archive. This would aid in the eventual goal of producing archival measurements of the stellar activity parameter $\log R'_{HK}$. Integral to this goal was developing a means of reliably achieving a good background subtraction of Échelle CCD spectra, specifically in the regions of the Ca II resonance lines.

The main focus of this work was the development of new software for this task, as well as testing that software against existing background subtraction methods. The results of Sections 13 and 14.1 demonstrate that our newly developed background subtraction software, SKYBM, is capable of producing accurate background subtractions in the region of the Ca II H and K lines. In these Sections it was also shown to be more accurate than the background subtraction of both Options 6 and 22, offered by the ECHOMOP package.

ECHOMOP's Option 6 was found to produce accurate subtractions of the background light level but introduced additional noise to the output spectra. The determination of $\log R'_{HK}$ (detailed in Section 4.2) is very sensitive to the measured level of flux in the Ca II H and K line cores. Therefore any unwanted noise in this region will have a negative impact on the accuracy of the calculation of this value. For this reason Option 6 might not be suitable for our investigations.

ECHOMOP's Option 22 was found to under-approximate the background light contribution in this region resulting in an insufficient background subtraction. Not being able to see internally into the operations of the ECHOMOP package it is difficult to investigate the cause of this. Suffice to say that accurate removal of the unwanted

background is vital to the calculation of $\log R'_{HK}$ and so Option 22 will also not be suitable for our investigations.

In Section 13.2 it was found that the SKYBM software was capable of producing the most accurate subtractions of the background in this region of interest. This comes at the cost of greater input and execution time. Additional attention must be paid to the creation of the accompanying Échelle order mask to ensure that only background data is subtracted. However, with the input of an accurate mask, the SKYBM software has been demonstrated to provide the best background subtraction of the three methods used in this work. It is therefore expected that this software will be produce suitably subtracted spectra for $\log R'_{HK}$ determination.

The calculation of $\log R'_{HK}$ requires spectra of high enough quality that the flux in the regions of the Ca II resonance lines can be accurately measured. In Section 14.1 we found that, for all subtraction methods, only the stellar spectra having SNR >100 produced usable quality results. It is therefore recommended that any future use of this pipeline only use spectra with SNR >100.

16. Future of these Investigations

In this Section we detail several key areas of this project where we believe it could be improved and expanded upon. As it stands the data reduction procedure that was developed is capable of working with any spectra from the SOPHIE archive and producing background subtracted collapsed Échelle spectra. However this process requires a lot of manual setup and input throughout, something which would amount to a considerable time investment if applied to the larger archive. Also the resulting processed spectra are unit-less, having not been subject to wavelength or flux calibration, steps which are essential for the subsequent calculation of log R'_{HK}.

Code Improvements

Flux and Wavelength Calibration

The ECHOMOP Échelle reduction procedure was not run to completion in this project. The ECHOMOP pipeline provides additional procedures to perform the necessary flux and wavelength calibrations on the collapsed spectra. These steps could easily be included by continuing with the guided options given by the echmenu tool.

log R'_{HK}

As discussed in Section 2 the stellar activity index log R'_{HK} is a useful tool in the hunt for mass-losing planets. This parameter has not been measured for the spectra in the SOPHIE archive. It would useful therefore to incorporate code to calculate this property, as described in Section 4.2, for each stellar spectrum into the data reduction procedure. Parameterising stellar spectra in this way will allow for the simple numerical comparison of different stars in a way similar to Figure 9 (plot of log R'_{HK} vs B-V). If nothing else the resulting log R'_{HK} data will be publishable.

Code efficiency

The SKYBM code is more computationally intensive than either of the ECHOMOP options. These are likely to be much more streamlined in their design being part of a published software package. Throughout my experience developing the code for this project there were many instances were code designed for a specific task was able to be simplified and reduced, producing the same result for less execution time. There are undoubtedly areas in the code which can be improved in this manner and hence improve the overall code efficiency and reduce computational burden.

Pipeline automation

Further to building on pre-existing code, opportunities exist to increase the automation of the data reduction process. More automation would mean that all reduction steps could be managed from a single master pipeline script which could then be run for the whole SOPHIE archive.

Data Acquisition

The process of selecting and obtaining night's spectra for a target was laborious as there were many reasons why target data would not be appropriate. Any chosen archival data must therefore be carefully inspected prior to download. A script could be created to supervise data acquisition using smart filters to identify and download only the appropriate night's spectra. For these investigations this would mean spectra of F, G and K main sequence stars, with SNR greater than 100 and observed in HR mode.

MASK creation

The Mask creation procedure required for the SKYBM software requires manual assistance to determine the spectral order regions to be masked. New code could be developed to detect these regions and reliably create an appropriate mask removing the need for manual input at this stage.

ECHOMOP AUTOMATION

ECHOMOP provides automation to many aspects of Échelle data reduction which have traditionally required substantial interactive assistance. In this project we have already used Starlink commands within scripts, this could also be extended to ECHOMOP processes to apply a standard reduction procedure to spectra.

Reducing the SOPHIE archive

As shown by the population histograms of Figure 13 and Figure 14, there are many F, G and K main sequence stars with thousands of recorded spectra in the SOPHIE archive. A significant number of potential targets increase the chances of finding other systems exhibiting similar spectral features to that of WASP-12, potentially indicating proximal evaporating planets. If such candidates are found that are magnitudes brighter than WASP-12 they would yield significantly higher quality spectra essential for the precise measurement and study of this phenomena. Brighter targets would also decrease the effects of background noise dominating the low flux regions of interest. In addition to higher quality spectra, brighter stars RV curve should be very easy to measure, for follow up investigations. Any identified planet systems would become high priority targets for follow up by the exoplanet community leading to a better understanding of exoplanet chemical composition and also provide important constraints on the formation and

evolution as well as giving insight towards the rocky-planet mass-radius-composition relationship. If this pipeline proves successful with application to the SOPHIE archive, with appropriate modification it could be applied to other Échelle spectra archives.

References & Bibliography

References

Ayres, 1979, ApJ, 228, 509
Bakos et al., 2007, ApJ, 671, L173
Baliunas S. et al., 1995, ApJ, 438, 269
Ballard S. et al., 2011, ApJ, 743, 2, 200
Boisse et al., 2010, A&A, 523, A88
Bouchy et al. 2008, A&A, 482, 25
Bouchy et al. 2009b, A&A, 505, 853
Bouchy et al., 2010, A&A, 519, A98
Cameron et al. 2007, MNRAS, 375, 951
Cassan A. et al., 2012, Nature, 481, 167-169
Delrez et al., 2014, A&A, 563, A143
Duncan et al., 1991, ApJ, 76, 383
Enoch et al., 2011, AJ, 142, 3
Fossati et al., 2010b, ApJL, 714, L222
Fossati et al., 2011, IAUS, 276, 163
Fossati L. et al., 2013b, ApJL, 766, L20
Fossati et al., 2014, ASS, 354, 21-28
Haswell et al., 2012, ApJ, 760, 79
Hebb et al., 2009, ApJ, Vol 693, 2, pp. 1920-1928
Hebrard G. et al., 2010, AA, 513, A69
Henry et al. 1996
Jenkins J. et al., 2015, ApJ 150, 56
Joshi Y. et al., 2009, MNRAS, 392, 1532
Kalas et al., 2013, ApJ, 775, 1, 2013
Kostov V. et al., 2014, ApJ, 784, 14
Knutson et al., 2010, ApJ, 720, 1569
Lanza, 2014, A&A, 572, L6
Liu M. et al, 2013, ApJL, 777, 2, L20
Mayor & Queloz 1995, Nature, 378, 355
Mordasini et al., 2009, A&A, 501, 1161
Noyes et al., 1984, ApJ, 279, 763
Perez-Becker and Chiang, 2013, MNRAS, 433, 2294
Rappaport et al., 2012, ApJ, 752, 1
Rogers L.A. & Seager S., 2010, ApJ, 712: 974R
Savitzky & Golay, Analytical Chem., 1964, 36, 1627
Seager et al. 2000, ApJ, 540, 504
Seager S.et al., 2007, ApJ, 669: 1279-1297
Steffen J. et al., 2013, MNRAS, 428, Issue 2, 1077-1087
Sumi, T. et al., 2011, Nature 473, 349–352
Vaughan et al., 1978, PASP, 90, 267
Wenger et al., 2000, A&AS, 143, 9
Wilson, 1978, ApJ, 226, 379
Wolszczan A. & Frail D., 1992, Nature, 355, 145
Wright 2004, AJ, 128, 1273

Wright et al. 2004, ApJ, 152, 261
Wright, J, T et al, 2012, ApJ, Vol 753, 2

Bibliography

Exoplanets.eu – Reference on specific exoplanets

An Introduction to Astrophysics, A. Norton – Background in astronomy and RV method

Transiting Exoplanets, C. Haswell – Background on Exoplanet research

Astronomical Spectroscopy, J. Tennyson – Background on Spectroscopy

Appendix 1: OHP COST STSM Report

Origin Action TD1308: Short Term Scientific Missions (STSM), Mission Report

Research Trip to Observatoire de Haute-Provence (OHP)

COST STSM Reference Number:	COST-STSM-TD1308-24985
STSM Candidate:	Calum McCune, The Open University, Milton Keynes (UK), calum.mccune@open.ac.uk
STSM Topic:	Understanding the formation and evolution of planetary systems and habitable planets
Host:	Auguste Le Van Suu, Observatoire de Haute-Provence, St.Michel/l'Observatoire(FR), mira.veron@osupytheas.fr
Observing dates:	27/04/15 – 01/05/15
Away dates:	26/04/15 – 02/05/15

Aim: To observe key northern hemisphere targets that have been selected as their anomalously low log R'_{HK} values may indicate the presence of stellar gas shrouds, and hence the presence of undiscovered extremely close-in planets, losing mass due to intense stellar irradiation.

Background: Our group's work on WASP-12b revealed that mass loss from short-period planets can produce a shroud of diffuse gas which engulfs the entire planetary system and produces noticeable absorption of the stellar spectrum in the resonance lines of abundant species [1,2]. By examining archival spectral data on bright stars we have identified about 50 which show signs of enshrouding gas [3]. Based on this, we are carrying out a precisely-targeted radial velocity planet search. This would enable improved characterisation and a search for time-variability of the circumstellar absorption. We proposed a targeted radial velocity planet search at high, sub-day, cadence, of suspected planet hosting stars, with the goal of finding important new planets. As these are bright stars, the discovery of evaporating planets would offer unprecedented opportunities to examine their chemical composition. A

better understanding of exoplanet chemical composition will also provide important constraints on the formation and evolution.

Any identified planet systems would become high priority targets for follow up by the exoplanet community. The observing strategy enables detection of short-period planets down to about 2 Earth masses in only 5 few nights of telescope time. SOPHIE is one of two accessible northern hemisphere instruments with the required RV precision to achieve this.

I am a PhD student working on the chemical characterisation of mass loss in exoplanetary systems, currently in my first year. In addition to the science goals of this mission, this COST Action was an invaluable experience for me, allowing me to gain hands-on experience collecting data relevant to my studies as well as gaining general transferable observing skills.

Methodology: We observed our targets over 5 nights from 27/04/15 – 01/04/15, in total 64 target observation exposures were made. All spectrograph functions are controlled via dedicated software. Using the New Short Term Scheduler (NSTS) the schedule of the nights observations were managed. A list of the night's targets were prepared in advance of arrival and uploaded to the NSTS for each night. The order of the schedule could be changed at any point during the night and parameters of science exposures were manually modified when necessary to account for changes in weather.

Prior to observations on each night, a series of calibration measurements and exposures were taken, as per the predefined list of SOPHIE standard calibrations. These include the flat-field and bias frames necessary for data reduction. Typically the most recent Th-Ar calibration frame is used to estimate the instrumental drift. The drift tracks the changes in the instrumental behaviour during the observing night. Depending on the extent of the drift rate, additional Th-Ar calibration exposures were added into the night's schedule to monitor the real-time drift.

Using the Data Reduction Software (DRS) at the telescope, the science exposures were subjected to an automatic cross-correlation process using the appropriate spectral template. All data reduction and re-reduction can only be carried out at the telescope since the software is not publically available. Further cross-correlation was carried out using the offline DRS to obtain the optimal radial velocity measurements since the DRS automatically chooses the spectral type template to use. Our stay coincided with the full moon, and hence for some target exposures, the solar spectrum light contamination had to be accounted for using the dedicated offline DRS routines.

Results: All observing nights were affected to different extents by weather conditions, namely clouds and occasional rain, which also resulted in variable atmospheric seeing conditions. This restricted our ability to obtain high SNR exposures and limited the number of observations we were able to perform. Nonetheless we were able to obtain useful data.

DATE	JD	TIME (UTC)	TARGET	SNR	RV (km/s)	ΔRV (km/s)	Centre of CCF
28/04/15	2457141.322	19:44:00	TARGET A	48.6	49.48863	0.0032	[49]
28/04/15	2457141.34	20:09:00	TARGET C	33.8	-40.3829	0.0074	
28/04/15	2457141.357	20:34:00	TARGET B	27.3	-1.51203	0.0044	[-1]
28/04/15	2457141.397	21:31:00	TARGET D	25.2	-77.79457	0.0046	[-78]
28/04/15	2457141.413	21:54:00	TARGET C	31.7	-40.40012	0.0083	
28/04/15	2457141.428	22:17:00	TARGET A	23.4	49.45467	0.009	[49]
28/04/15	2457141.448	22:45:00	TARGET B	15.8	-1.57029	0.0089	[-1]
28/04/15	2457141.468	23:14:00	TARGET A	68.1	49.49523	0.0021	[49]
28/04/15	2457141.488	23:42:00	TARGET B	61.7	-1.48104	0.0019	[-1]
29/04/15	2457141.508	00:12:00	TARGET D	45.9	-77.7801	0.0024	[-78]
29/04/15	2457141.528	00:41:00	TARGET C	61.3	-40.38775	0.0039	
29/04/15	2457141.549	01:10:00	TARGET A	94.1	49.4999	0.0015	[49]
29/04/15	2457141.569	01:40:00	TARGET B	58.8	-1.48612	0.002	[-1]
29/04/15	2457141.594	02:15:00	TARGET D	35.1	-77.7869	0.0032	[-78]
29/04/15	2457141.608	02:35:00	TARGET E	196.1	0.68787	0.0015	
29/04/15	2457141.628	03:05:00	TARGET B	43.5	-1.49585	0.0028	[-1]
29/04/15	2457141.648	03:33:00	TARGET A	84.9	49.48078	0.0018	[49]
29/04/15	2457141.658	03:47:00	TARGET E	232.9	0.69227	0.0013	

Table 5: SOPHIE RV Data, Night 2 (30/04/2015)

Table 5 above shows all of the science RV data obtain on Night 2 (30/04/2015). I have chosen to show this night as it had the most favorable weather conditions and hence the highest SNR data of all the nights. All data shown has been cross-correlated with the mask, or sky background subtraction, appropriate for each target.

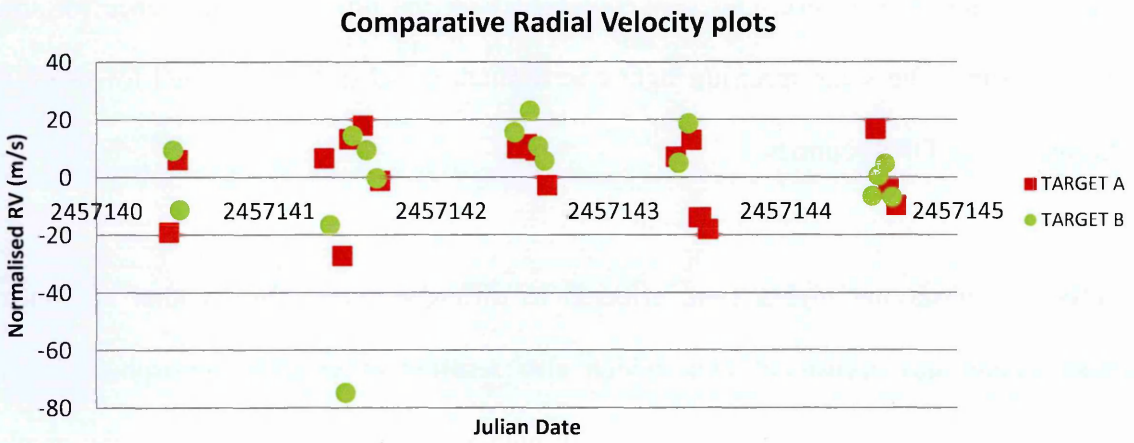


Figure 39: Normalised RV data for two best targets

Figure 38 shows the RV data for all nights for the two most successful targets, in terms of number of exposures. The respective radial velocities of each target are shown with respect to the mean so that they can be displayed on the same plot.

Discussion: From the preliminary data shown in Figure 38, it is not yet clear whether the radial velocities show significant trends that may be due to a planet. There is still the need to correct for other factors such as SNR variation, instrumental drift and atmospheric contributions. Continued analysis to identify and remove trends in the data is currently ongoing.

References: [1] Haswell C. A. et al. 2012. *ApJ.* 760:79H. [2] Fossati L. et al 2013. *ApJL.* 766:L20. [3] Staab D. et al. 2015 in prep.

Appendix 2: Starfinder.py Code (Python)

```
# Name:
#     starfinder
#
# Purpose:
#     Script opens a text file containing a list of object identifiers,
#     removes duplicate names, removes non-stellar object names and
#     finally alphabetizes the new list.
#
# Operation:
#     A copy of this script must be saved in the same directory as the
#     object list text file and the text file 'remove_list.txt'. The user
#     is prompted for the name of the object list file, the file contents
#     are then extracted and treated by this scripts functions. First the
#     function 'starfinder()' removes any object names which are repeated
#     within the file, leaving a list of unique object names. Next the
#     function 'remover()' removes any non-stellar object names by
#     referencing the text file 'remove_list.txt'. This file contains a
#     list of non-stellar objects whose spectra exist in the SOPHIE
#     archive. Finally the function 'alpha_abs()' alphabetises the
#     remaining list. Thus leaving a list of the unique stellar objects
#     from the original list.
#
# Author:
#     Calum McCune, 2015 (Open University)


###### FUNCTIONS ##################

def starfinder(filename):       # function to remove duplicate object
                                # names from the input list text file

      # opens input file as designated by user in main(). Extracts file
      # data to the list 'file_contents'.
      input_file = open(filename, 'r')
      file_contents = input_file.readlines()
      input_file.close()

      output_file = open(filename, 'w')   # file re-opened to hold new
                                          # list data

      count = 0                   # integer value used to count no. of
                                  # object names contained within the list
                                  # file.
      true_count = 0              # integer value used to count no. of
                                  # unique object names contained within the
                                  # list file

      for word in file_contents: # for loop counts no. of objects in file
          count += 1

      unique_words = set(file_contents) # function set() used to create
                                  # the list 'unique_words' containing the
                                  # names of all objects mentioned in the
                                  # file.

      for word in unique_words:  # for loop outputs the contents of the
                                  # list 'unique_words' to the output file
                                  # and updates the value of true_count for
                                  # each.
          output_file.write(str(word))
          true_count += 1

      print 'number of objects found %d' % count
      print 'number of unique objects found %d' % true_count
```

```
    output_file.close()


def starcount(filename):        # function to count the numer of objects
                                # in a list file. Used in the development
                                # of this script but not in the execution.
                                # This function may be of use when
                                # adjusting this script for other spectral
                                # archives.
    total = 0

    star_file = open(filename)

    for line in star_file:
        total += 1

    star_file.close()

    print 'No. of objects in file: ',total


def remover(filename):          # function to remove all non-stellar
                                # object names contained in a list of
                                # spectra from the SOPHIE archive.
    '''function opens a text file and removes all phrases that are
    contained in the file 'remove_list.txt'
    '''

    # opens the input list file. Extracts file data to the list
    # 'start_list'.
    start_file = open(filename)
    start_list = start_file.readlines()
    start_file.close()

    # opens the list file 'remove_list.txt'. Extracts file data to the
    # list remove_list
    remove_file = open('remove_list.txt')
    remove_list = remove_file.readlines()
    remove_file.close()

    keep_file = open(filename,'w') # reopens object list file for writing

    for line in start_list:     # for loop write only the lines in
                                # 'start_list' which do not exist in
                                # 'remove_list'. Thereby removing all non-
                                # stellar objects from the object list
                                # file.
        if line not in remove_list:
            keep_file.write(line)

    print 'Non-stellar objects removed'

    keep_file.close()


def alpha_abs(filename):        # function alphabetises list file
    '''function opens a text file and re-orders the contents line by line
    alphabetically.
    The contents of this input file are replaced with the alphabetically
    re-ordered contents'''
    alpha_File = open(filename, "r")

    outstr = ''

    for word in sorted(alpha_File, key = str.lower):
        outstr += word

    alpha_File.close()
```

```
    alpha_File = open(filename, "w")
    alpha_File.write(outstr)


    alpha_File.close()

##################################


def main():                         # main function of the script
    ### USER INTRODUCTION ###
    print 'STARFINDER PROGRAM\n'

    # prompts user for the input filename. As a failsafe amends the file
    # input string if the file suffix .txt is not included.
    filename = raw_input('\nPlease enter the object list file name: ')
    if '.txt' not in filename:
        filename += '.txt'

    print 'filename: %s' %filename
    raw_input('\nPress any key to continue...\n\n') # awaits any user
                                    # input to continue with the program.

    # functions executed for the given input file name
    starfinder(filename)
    remover(filename)
    alpha_abs(filename)


main()
```

Appendix 3: CCD Frame Preparation Scripts (cshell)

3.1 makelog

```
!/bin/tcsh


#

#  Name:

#     makelog

#

#  Purpose:

#     [First] script in the pre-echomop reduction process. Script creates
#     a log file containing the information of the contents of each file
#     as read from the file header.
#
#  Operation:

#     Utilising functions 'FITSHEAD' and 'grep' this script reads through
#     the FITS header information of each file, extracting parameters
#     important for creating the pre-ECHOMOP frames and storing this
#     information in the log file 'sophie.log'. Information recorded in
#     this log file: filename, target name (if applicable),
observation
#     type (i.e. BIAS, WAVE, STAR, LAMP), date, time, exposure time,

#     mode (HE or HR). This log file will be referenced by later scripts
#     which operate only
 on specific frame types.

#

#  Usage:

#     This function is called within 'tweek_pipeline' for each nights
#     data folder

#

# Author:

#     John Barnes (Open University)




###### LIBRARIES ######

star32            # initialises starlink, needed for figaro and kappa

figaro            # initialises figaro

kappa       # initialises kappa

#######################
```

```
echo >! sophie.log
\rm -f junk.fits.bz2 junk.fits

foreach file0 ( *.fits )


    \cp -f $file0 junk.fits

    set file = junk.fits


    fitshead $file >! file.dat

    set value  = `cat file.dat | grep -i "TARG NAME    =" | sed s:\':\ :g
| sed s:\/:\ :g`

    set object = `echo $value[6]$value[7] | sed s:\':\ :g`


    set value  = `cat file.dat | grep -i "DPR TYPE" | sed s:\':\ :g | sed
s:\/:\ :g | sed s:\,:\ :g`

    set otype = `echo $value[6] | sed s:\':\ :g`

    echo OTYPE: $value


    set value  = `cat file.dat | grep -i "DATE" | sed s/\'/\ /g | sed
s:\/:\ :g | sed s/T/\ /g`

    set dateobs = `echo $value[4] `

    set timeobs = `echo $value[5] `

    set value  = `cat file.dat | grep -i "CCD UIT"`

    echo UIT : $value


    set exptime = `echo $value[6] | sed s:\':\ :g | sed s:\/:\ :g`

    echo EXPTIME : $exptime


  set value  = `cat file.dat | grep -i "INS FIB" | sed s:\':\ :g | sed
s:\/:\ :g | sed s:\(:\ :g | sed s:\):\ :g `

    echo MODE : $value

    if ( $value[1] == "Binary" ) then

       set mode = $value[1]

    else

       set mode = $value[6]

    endif

    echo MODE : $mode
```

```
    echo "$file0 \t\t $object \t\t $otype \t\t  $dateobs \t\t $timeobs
\t\t $exptime \t\t $mode"

    echo "$file0 \t\t $object \t\t $otype \t\t  $dateobs \t\t $timeobs
\t\t $exptime \t\t $mode" | awk '{printf
"%30s %25s %25s %15s %10s %10s %10s  \n", $1,$2,$3,$4,$5,$6,$7}' >>
sophie.log


end
```

3.2 make_ndfs

```
#!/bin/tcsh


#

#  Name:

#     make_ndfs

#

#  Purpose:

#     [Second] script in the pre-echomop reduction process.

#     This script renames all FITS files are creates copies in NDF format.

#

#  Operation:

#     Script loops through each FITS file in the current directory and
#     changes the syntax of specific symbols in the filenames which
#     starlink has issues reading. Copies of these renamed files in FITS
#     (.fits) format are created converted to starlinks native NDF

#     (.sdf) format.

#

#  Usage:

#     This function is called within 'tweek_pipeline' for each nights
#     data folder

#

#  Author:

#    John Barnes (Open University)




###### LIBRARIES ######

star32            # initialises starlink, needed for figaro and kappa

figaro            # initialises figaro
```

```
kappa       # initialises kappa, needed for convert
convert           # initialises convert, needed for fits2ndf function
######################



foreach file ( SOPHIE*.fits )       # loop runs for every .fits file in
                                # current directory
 with the prefix
                                # "SOPHIE". As is the case for all
                                # SOPHIE archival files.
  set fileout = `echo $file:r | sed 's/\./\_/;s/\.//;s/\://g'`    #
declares
                                # the name of the
 resulting output
                                # to be the same as that of the
                                # input
 only where the 'sed'
                                # function replaces the
 problematic
                                # symbols with others more amenable
                                # to
 starlink.


  echo mv $file $fileout'.fits'           # status check: prints to the
                                # terminal letting the
user know
                                # the following code has been
                                # executed
       mv $file $fileout'.fits'           # renames the input FITS file to
                                # the starlink friendly
 output
                                # filename declared above as
                                # 'fileout'

  echo fits2ndf $fileout'.fits' $fileout # status check: prints to the
                                # terminal letting the user know
                                # the following code has been
                                # executed

       fits2ndf $fileout'.fits' $fileout # creates an NDF copy of the
                                # renamed input FITS file


End
```

3.3 preprun_master

```
#!/bin/tcsh


#
#  Name:
```

```
#     preprun_master
#
#  Purpose:
#     [Third] Master script which calls all remaining scripts in the
#     pre-echomop reduction process. Sorts image files by observation
#     type ready for subsequent scripts.
#
#  Operation:
#     Script begins by creating the log file 'sophie.logHR' which
#     contains only the HR mode file header information from
#     'sophie.log'. This is required as all subsequent scripts are
#     designed to only work with HR mode files. Data files
#     'tempbias.dat', 'tempflat.dat',     'temparc.dat' and 'tempobj.dat'
#     created containing the names of files of each respective
#     observation type as read from 'sophie.logHR'. Runs the script
#     'preprun_ext1_auto' which manges the execution of remaining scripts.
#
#  Usage:
#     This function is called within 'tweek_pipeline' for each nights
#     data folder
#
#  Authors:
#     John Barnes (Open University)


   grep -v " HE " sophie.log >! sophie.logHR    # creates the new log file
                        # 'sophie.logHR' containing only file
                        # information from 'sophie.log' for HR mode
                        # files. This is because the remaining
                        # scripts are only designed to run for HR
                        # mode.

   if ( ! -d Reduce ) then    # if statement creates the directory
                        # 'Reduce' if it does not exist already.
```

```
    mkdir Reduce        # This will be used to hold all resulting
  endif                 # reduced data


setenv Dir `echo $cwd`        # sets the string 'Dir' to be the pathway
                        # of the current directory

echo ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
echo BASE DIRECTORY IS $Dir
echo ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

echo "CURRENTDATE: $CURRENTDATE"    # status check

  grep BIAS sophie.logHR | cut -c 1-40 >! tempbias.dat # creates the data
                            # file 'tempbias.dat' containing
                            # just the file names of
                            # observation type BIAS, as
                            # identified from the
                            #'sophie.logHR' log file. These are
                            # the 'bias frames'
  echo BIAS : `cat tempbias.dat` \\n     # outputs all file contents to
                            # terminal window which can then be
                            # checked by the user

  grep LAMP sophie.logHR | cut -c 1-40 >! tempflat.dat # creates the data
                            # file 'tempflat.dat' containing
                            # just the file names of
                            # observation type LAMP, as
                            # identified from the
                            # 'sophie.logHR' log file
  echo FLAT : `cat tempflat.dat` \\n     # outputs all file contents to
                            # terminal window which can then be
                            # checked by the user

  grep WAVE sophie.logHR | cut -c 1-40 >! temparc.dat # creates the data
```

```
                              # file 'tempbias.dat' containing
                              # just the file names of
                              # observation type BIAS, as
                              # identified from the
                              # 'sophie.logHR' log file
   echo WAVE : `cat temparc.dat` \\n      # outputs all file contents to
                              # terminal window which can then be
                              # checked by the user

grep STAR sophie.logHR | cut -c 1-40 >! tempobj.dat # creates the data
                              # file 'tempbias.dat' containing
                              # just the file names of
                              # observation type BIAS, as
                              # identified from the
                              # 'sophie.logHR' log file
   echo OBJ  : `cat tempobj.dat`  \\n     # outputs all file contents to
                              # terminal window which can then be
                              # checked by the user


   echo
 ------------------------------------------------------------------------
-------------------------------
   echo

  ./preprun_ext1_auto               # runs the script
                             #'preprun_ext1_auto' which runs all
                              # of the remaining scripts

   @ i ++
```

Appendix 4: ECHOMOP Preparation Scripts (cshell)

4.1 preprun_ext1_auto

```
#!/bin/tcsh

#+
#  Name:
#     preprun_ext1_auto

#  Purpose:
#     Driver script to set up for an automated echelle data reduction run.

#  Language:
#     C shell script.

#  Description:
#     This script can be used to coordinate the relatively complex
#     series of operations required for reducing a large number of
#     similar echelle spectrograms.  Before using the script, you
#     should be familiar with the use and parameters of the ECHOMOP
#     package.  (You can try this out without being familiar with
#     ECHOMOP - but it is a complex package!)
#
#     Essentially the procedure is to "prototype" the reduction
#     manually using ECHOMOP and then, once you have determined
#     suitable parameter settings, to use the manually-generated
#     reduction structure file as a template with which to reduce
#     the complete dataset.
#
#     Which parts of the reduction procedure need to be done for every
#     frame, and which parts can be copied "as-is" from the manual
#     template will depend on your data.  For example, you might
#     use the order traces from the manual reduction for all the
#     frames.  This will be fine as long as the image of the echellogram
#     remains stable over the full time period which covers your dataset.
```

```
#     One way to check this sort of thing is to perform two manual
#     reductions - one from early in the time period covered, one from
#     late - and then compare the two.  Plotting orders from the reduced
#     arcs is a good way to spot shifts in the dispersion direction.
#     Detecting shifts in the spatial direction can be more difficult;
#     however, you might use the "tracepoly" script to extract the
#     parameters of the order traces from the ECHOMOP reduction structure
#     files.  You can then compare the parameters and look for shifts -
#     for POLY fits checking that the low-order parameters closely
#     match and that higher-order parameters are small should be enough.
#
#     This script calls a set of shortish scripts to perform each of the
#     data preparation tasks - debiasing, flat fielding, clipping and so
on.
#     You should review the descriptions in these scripts so that you are
#     happy you understand what each one is doing with your data.  You may
#     also need to edit the scripts in some places, particularly if your
#     echellograms have orders which run roughly vertically.
#
#     The output of the automated reduction process is a series of
#     files ob_"file" where "file" is the name of the source object frame.
#     Arc frames ar_"file" are similarly created.

#  Usage:
#     You need to know the true detection area of the CCD used to acquire
#     your data - if you do not, display an image with FIGARO IMAGE and
#     look for empty parts of the frame at the edges of the image.
#     These are not used by ECHOMOP and should be cut off by setting
#     suitable values for the trim parameters.  You should select a part
#     of the overscan ("dark" area) to be used for measurement of the
#     electronic bias level.  Use FIGARO ICUR to measure the coordinates
#     of the various areas.
#
#     The following in this script should be edited to suit your data:
#
```

```
#       1. Detector overscan sample region - for bias-level
determination.
#       2. Detector clipping region (to remove overscan) - to remove
#          non-science data areas of the input images.
#       3. Detector output details (noise, gain).
#       4. List of bias frames.
#       5. List of flat-field frames.
#       6. List of arc frames.
#       7. List of arc mask frames (paired with arcs) - these are used
#          to configure the processing of arcs so that they are extracted
#          in the same way as objects.
#       8. List of object frames.
#       9. Name of prototype ECHOMOP reduction structure file.
#
#    See comments in the script for details.  Example values
#    have been given for some of these items.

# Arguments:
#    This script does not use any command-line arguments.

# Notes:
#    1. The command line
#
#          % nohup source preprun2 &
#
#       will ensure that the script continues to run even when you
#       have logged off the system.  The "&" at the end of the line
#       will run the script in the background.
#
#    2. The environment variables:
#
#        $xbimin
#          X-start of overscan region to use for bias subtract.
#        $xbimax
#          X-end of overscan region to use for bias subtract.
```

```
#         $ybimin
#           Y-start of overscan region to use for bias subtract.
#         $ybimax
#           Y-end of overscan region to use for bias subtract.
#         $xtrmin
#           X-start of region of image to be retained.
#         $xtrmax
#           X-end of region of image to be retained.
#         $ytrmin
#           Y-start of region of image to be retained.
#         $ytrmax
#           Y-end of region of image to be retained.
#         $Gain
#           Detector output characteristic (e/ADU).
#         $RDN
#           Detector output noise (e).
#         $EchFile
#           Name of the ECHOMOP reduction structure file to be used.
#
#        are SET by this script.
#
#     3. This script MUST be sourced to work at all!
#
#     4. The following scripts are called by this master script.
#        They should be present in the working directory.
#
#         echrdarc - reduces an individual arc frame using ECHOMOP.
#         echrduce - reduces a series of object frames using ECHOMOP.
#         preparcs - uses FIGARO to prepare a series of arc frames
#                    by debiasing, flat fielding and so on.
#         prepbias - uses FIGARO to merge a set of bias frames into a
median
#                    frame.
#         prepflat - uses FIGARO to merge a set of flat-field frames into
#                    a median frame.
```

```
#         prepobjs - uses FIGARO to prepare a series of object frames
#                    by debiasing, flat fielding and so on.
#
#     5. The called scripts will work with FIGARO v5.0-0 or later.

#  Authors:
#     MJC: Martin Clayton (Starlink)
#     CSM: Calum McCune (Open University)
#
#  History:
#     06-NOV-1995 (MJC):
#       Original Version.
#     JUN-2015 (CSM):
#     Minor edits made so that script can run within pipeline and with
#     SOPHIE archival data
#-

#  (Bias and trim parameters are for EEV chip on UCLES unbinned
#  in the spatial direction.)


#  Overscan sample region.
      setenv xbimin 15;
      setenv xbimax 50;
      setenv ybimin 23;
      setenv ybimax 4100;

#  Clipping region.
      setenv xtrmin    54;
      setenv xtrmax  2101;

      setenv ytrmin   11.;
      setenv ytrmax  4087.;
```

```
#  Display the regions selected.
      echo '';
      echo "Bias region in X: $xbimin to $xbimax";
      echo "            in Y: $ybimin to $ybimax";

      echo '';
      echo "Clipping region in X: $xtrmin to $xtrmax";
      echo "                in Y: $ytrmin to $ytrmax";


# Define reduction directory

      setenv RedDir $Dir/Reduce
      echo "Reduce folder: $RedDir"

      setenv CCD  eev         # string holds the manufacturer of the CCD
                        # to be appended
      setenv extn ext1


#  Detector output details - these are suitable defaults if you do not
#  the correct values.  You will have to determine the correct values if
#  you intend to use the optimal extraction algorithm or you require
#  variance estimates for the extraction.

      setenv RDN  6.0;
      setenv Gain 2.85714285;


#  Display detector details.
      echo '';
      echo " CCD = $CCD ", Extension = $extn;
      echo "CCD Readout noise: $RDN electrons";
      echo "CCD Output Gain:   $Gain electrons/ADU";
```

```
#-----------------------------
#  Location of raw bias frames

     setenv RawDir `echo $cwd`
      cd $RawDir

      setenv Date $CURRENTDATE

set BiasFrames = ( `cat tempbias.dat` )



#     Go to reduction directory to process bias frames
      echo \\n\\n " List of $#BiasFrames bias frames: $BiasFrames"  ;
echo " Processing bias frames in $cwd"


  ./prepbias $BiasFrames;



      echo
     echo
------------------------------------------------------------------------
      cd Reduce
      set NewBias = `cat current_bias.dat`
     cd ..
      setenv Bias $NewBias
    echo "NEW BIAS FRAME is $Bias"
     echo
------------------------------------------------------------------------\\n


#-----------------------------
#  Location of raw flatfield frames

      setenv RawDir $cwd
```

```
      cd $RawDir
      setenv Date $CURRENTDATE
      echo "Date working: $Date"
set FlatFrames = ( `cat tempflat.dat` )

#  Go back to reduction directory to process bias frames
      echo \\n\\n " List of $#FlatFrames flatfield frames: $FlatFrames"  ;
      echo " Processing flat frames in $cwd" ;

      ./prepflat $FlatFrames;

#------------------------------
#  Location of raw object frames

      setenv RawDir $cwd
      cd $RawDir
      setenv Date $CURRENTDATE
      echo "Date working: $Date"
#  List of object frames.

      set ObjFrames = ( `cat tempobj.dat` )

#  Go back to reduction directory to process object frames
      echo \\n\\n " List of $#ObjFrames object frames: $ObjFrames"  ;
echo " Processing object frames in $cwd"

     ./prepobjs $ObjFrames;

#------------------------------
#  Location of raw arc frames
      setenv RawDir $cwd
      cd $RawDir
      setenv Date $CURRENTDATE
```

```
      echo "Date working: $Date"
#  List of object frames.
      set ArcFrames = ( `cat temparc.dat` )

#  Go back to reduction directory to process ThAr frames
      echo \\n\\n " List of $#ArcFrames Arc frames: $ArcFrames"  ;  echo "
Processing object frames in $cwd"

      ./preparcs $ArcFrames;

#  End-of-file.
```

4.2 prepbias

```
#!/bin/tcsh

#+
#  Name:
#     prepbias

#  Purpose:
#     Script to prepare a single biasframe from a series of frames.

#  Language:
#     C shell script.

#  Description:
#     This script produces a single median image from a series of
#     "raw" CCD bias frames.  The median bias frame is created using
#     FIGARO MEDSKY.  The output image is called "biasfram", this
#     can be altered by editing the appropriate line in the script.

#  Usage:
#     This script can simply be invoked from the shell; in this case
#     the script will prompt for a list of the input bias images.
#     Alternatively, the list of bias frames can be supplied on the
```

```
#     command line, for example:
#
#        % prepbias run0800 run0801 run0802 run0856 run0857 run0858
#
#     If wildcarding facilities are available in your shell, you can use
#     them to simplify the command line, for example, the above would
#     become:
#
#        % prepbias run080[012] run085[678]
#
#     in the C shell.  This wildcarding facility is available when the
#     script prompts for a list of input images.

#  Notes:
#     1. If needed, the input parameters can be input at the command
#        line thus:
#
#           % nohup prepbias filename [filename...] &
#
#        the "nohup" command will ensure that the script continues
#        to run even when you have logged off the system.  The "&" at
#        the end of the line will run the script in the background.
#
#     2. This script is designed to be used as part of an automated
#        echelle data reduction package.  If you intend to use it
#        for this purpose, you should not change the name of the output
#        median bias frame "biasfram".  See the comments in the script
#        for changes which can be made if it is to be used stand-alone.
#
#     3. This script will work with FIGARO v5.0-0 or later.
#
#     4. This script is designed to be called by a master reduction
#        script.  See the example scripts "preprun1" and "preprun2"
#        for details.
```

```
#  Authors:
#     ACC: Andrew Collier Cameron (St. Andrews)
#     MJC: Martin Clayton (Starlink)
#     CSM: Calum McCune (Open University)

#  History:
#     ??-???-???? (ACC):
#       Original Version.
#     06-NOV-1995 (MJC):
#       Minor tidying up for Cookbook Version.
#     JUN-2015 (CSM):
#     Minor edits made so that script can run within pipeline and
#     with SOPHIE archival data

#-

#  Do Starlink login and set up for Starlink applications used.
#  You can comment out these lines if you already have these set up AND
#  you invoke the script by "source"ing it, for example:
#
#     % source prepbias run080[012] run085[678]
#
      #source /star/etc/cshrc;
      #source /star/etc/login;
      star32
      alias echo 'echo >/dev/null';
      setenv ADAM_USER $RedDir/prepbias_adam
      \rm -rf $ADAM_USER
      mkdir $ADAM_USER
      figaro;
      kappa;
      convert
      unalias echo;
```

```
#  Force standard UNIX commands to "normal" behaviour.
#  This is to remove any special alias for the remove command.
      set rm_old = `alias rm`;
      unalias rm;

#  Get the list of input bias frames, either from the command line
#  or by prompting.
      if ( "$#argv" == 0 ) then
         echo '';
         echo -n '? Bias Frame(s) > ';
         set Files = ( `echo $< ` );
      else
         set Files = ( $* );
      endif



#  Set the name of the output median bias frame.
#  You can change this line if you are using the script on its own.

  set Biasframe = 'biasfram_'$CCD'_'$Date;

#  CREATE NEW BIAS IF IT DOES NOT ALREADY EXIST

  if ( -e $Biasframe'.sdf' ) then

      echo \\n\\n "++++++++" $Biasframe".sdf has already been created for
$CURRENTDATE - skipping ""++++++++"

else

#  Create a new Bias Frame list file.
#  This file is used to hold the list of bias frames to be processed
#  by FIGARO MEDSKY.
```

```
      set Biasname = 'biaslist.dat'$CCD;
      if ( -e $Biasname ) then
         echo "! Removing old $Biasname";
         rm $Biasname;
      endif
      touch $Biasname;


#  Go through the list of files, checking whether they exist and writing
#  their names into the Bias Frame list file.
      @ found = 0;
      foreach File ( $Files )


echo      cp $RawDir/$File $RedDir
          \cp -f $RawDir/$File $RedDir ;
          \cp -f $RawDir/$File:r'.sdf' $RedDir
#          chmod u+rwx $RedDir/$File ;
#          bunzip2 -f $RedDir/$File ;

          cd $RedDir ;
#       makendfs $File $CCD
#       \rm -f $File
#          set FitFile = $File:r ;

          set RawFile = $File:r ;

          echo \\n-------------------------------------------------------
----
          echo Now working with $RawFile'.sdf' ;
#       echo "fitsdin files=$FitFile out=$RawFile fmtcnv"
#          fitsdin files=$FitFile out=$RawFile fmtcnv ;

          if ( -e $RawFile.sdf ) then
#             fits2ndf $File $RawFile
```

```
            echo $RawFile >> $Biasname;
            @ found++;

         else
            echo "! Could not find $RawFile.";
         endif
      end

#  Display count of Bias Frames found.
      echo "! $found bias frames found and written to $Biasname";

#  Run FIGARO MEDSKY to generate the bias frame.
      if ( $found != 0 ) then
         medsky $Biasname noscaled $Biasframe;
      endif
      echo "! Median bias frame written to $Biasframe.";

   endif


  echo $Biasframe >! current_bias.dat



#  Reset any user command aliases.
   alias rm $rm_old;

   echo \\n\\n  --- EXITING PREPBIAS --- \\n\\n

#  End-of-file.
```

4.3 prepflat

```
#!/bin/tcsh
```

```
#+
#  Name:
#     prepflat

#  Purpose:
#     Script to prepare a median flat-field for use by ECHOMOP.

#  Language:
#     C shell script.

#  Description:
#     This script takes a list of "raw" CCD flat-field frames and
#     produces a single median flat-field image suitable for use as
#     the FFIELD file in ECHOMOP.
#
#     Be advised that flat-fielding in echelle data reduction is not
#     easy - sometimes it's not even possible.  Refer to the
#     "Introduction to Echelle Spectroscopy" (SG/9) if in any doubt.
#
#     The script performs the basic CCD data processing steps of bias
#     subtraction and trimming.  Bias subtraction removes the zero-point
#     bias introduced by the camera electronics.  Trimming removes the
#     pre-scan and over-scan regions of the CCD image which contain no
#     science data and can confuse the algorithms in echelle data
#     reduction software.
#
#     If required, the CCD frames can be rotated so that the dispersion
#     direction of the echelle orders runs horizontally as required by
#     ECHOMOP.
#
#     The median flat-field is calculated using FIGARO MEDSKY.

#  Usage:
#     This script can simply be invoked from the shell; in this case
#     the script will prompt for a list of the input flat-field images.
```

```
#     Alternatively, the list of frames can be supplied on the
#     command line, for example:
#
#        % prepflat run0800 run0801 run0802 run0856 run0857 run0858
#
#     If wildcarding facilities are available in your shell, you can use
#     them to simplify the command line, for example, the above would
#     become:
#
#        % prepflat run080[012] run085[678]
#
#     in the C shell.  This wildcarding facility is available when the
#     script prompts for a list of input images.
#
#     In practice, invocation from your shell is unlikely to be a good
#     method of using this script as 8 environment variables defining
#     the region of the image to be kept and the region of the overscan
#     to be used for debiasing are required.  Use of these variables is
#     summarised below.

#  Notes:
#     1. If needed, the input parameters can be input at the command
#        line thus:
#
#           % nohup prepflat filename [filename...] &
#
#        the "nohup" command will ensure that the script continues
#        to run even when you have logged off the system.  The "&" at
#        the end of the line will run the script in the background.
#
#     2. This script is designed to be used as part of an automated
#        echelle data reduction package.  If you intend to use it
#        for this purpose, you should not change the name of the output
#        median flat-field frame "fltfield".  See the comments in the
```

```
#       script for changes which can be made if it is to be used
#       stand-alone.
#
#    3. This script will work with FIGARO v5.0-0 or later.
#
#    4. When this script is invoked, 8 environment variables defining
#       the overscan region to be used for debiasing, and the region
#       of the images containing science data must be defined.
#       These environment variables are used:
#
#        $xbimin
#          X-start of overscan region to use for bias subtract.
#        $xbimax
#          X-end of overscan region to use for bias subtract.
#        $ybimin
#          Y-start of overscan region to use for bias subtract.
#        $ybimax
#          Y-end of overscan region to use for bias subtract.
#        $xtrmin
#          X-start of region of image to be retained.
#        $xtrmax
#          X-end of region of image to be retained.
#        $ytrmin
#          Y-start of region of image to be retained.
#        $ytrmax
#          Y-end of region of image to be retained.
#
#    5. A file "biasfram" containing a bias frame prepared by the
#       script "prepbias" should exist in the working directory.
#       You can alter the name of this file, see comments in the
#       script.
#
#    6. The input frames are NOT rotated by this script.  You may
#       have images in which the orders run roughly vertically,
#       in which case you should uncomment the line using FIGARO
```

```
#        IROT90 as indicated in the script.  Approximately horizontal
#        orders are required by ECHOMOP.  If you do rotate the flat
#        field, note that the script only rotates the final median
#        frame, not the individual input frames (saves time).
#
#     7. This script is designed to be called by a master reduction
#        script.  See the example scripts "preprun1" and "preprun2"
#        for details.

#  Authors:
#     ACC: Andrew Collier Cameron (St. Andrews)
#     MJC: Martin Clayton (Starlink)
#     CSM: Calum McCune (Open University)

#  History:
#     ??-???-???? (ACC):
#       Original Version.
#     06-NOV-1995 (MJC):
#       Minor tidying up for Cookbook Version.
#     JUN-2015 (CSM):
#     Minor edits made so that script can run within pipeline and
#     with SOPHIE archival data

#-

#  Do Starlink login and set up for Starlink applications used.
#  You can comment out these lines if you already have these set up AND
#  you invoke the script by "source"ing it, for example:
#
#     % source prepflat run080[012] run085[678]
#


      star32
      alias echo 'echo >/dev/null';
```

```
      setenv ADAM_USER prepflat_adam
      \rm -rf $ADAM_USER
      mkdir $ADAM_USER
      figaro;
      kappa;
      hdstools;
      unalias echo;

#  Force standard UNIX commands to "normal" behaviour.
#  This is to remove any special alias for the remove command.
      set rm_old = `alias rm`;
      unalias rm;

#  This line should only be modified if you wish to run several jobs
#  simultaneously, or you keep your Starlink parameter files in a
#  non-default directory.  If you do not understand this then leave the
#  line as it is!
      set ADAMDIR = $ADAM_USER;

#  Get the list of input flat-field frames, either from the command line
#  or by prompting.
      if ( "$#argv" == 0 ) then
         echo '';
         echo -n '? Flat-field Frame(s) > ';
         set Files = ( `echo $< ` );

      else
         set Files = ( $* );
      endif

#  Display information confirming the selected bias-level calculation
#  region.
      echo " Bias statistics will be calculated over X-range "\
 "$xbimin to $xbimax ";
      echo "                                  and over Y-range "\
 "$ybimin to $ybimax ";
```

```
#  Display information confirming the selected region of the image to be
#  retained; i.e., the CCD area containing the science data.
      echo " Frames will be trimmed over X-range $xtrmin to $xtrmax ";
      echo "                    and over Y-range $ytrmin to $ytrmax ";

#  Set script-internal variables to describe the trimming region.
      set Xmin = $xtrmin;
      set Xmax = $xtrmax;
      set Ymin = $ytrmin;
      set Ymax = $ytrmax;

#  Set the name of the input median bias frame.
#  You can change this line if you are using the script on its own.
#      set Biasframe = 'biasfram_'$CCD'_'$Date'_'$extn;
      set Biasframe = $Bias;

#  Set the name of the output median flat-field frame.
#  You can change this line if you are using the script on it's own.
      set Flatfield = 'fltfield_'$CCD'_'$Date;
      set BalanceFr = 'balframe_'$CCD'_'$Date;

   if ( -e $Flatfield'.sdf' ) then

      echo \\n\\n "++++++++++" $Flatfield'.sdf' "has already been created
for $CURRENTDATE - skipping ""+++++++++"

   else


echo PREPFLAT : "BIAS      = " $Biasframe
      echo PREPFLAT : "FLATFIELD = " $Flatfield
      echo PREPFLAT : "BALANCEFR = " $BalanceFr
```

```
#  Create a new flat-field frame list file.
#  This file is used to hold the list of frames to be processed
#  by FIGARO MEDSKY.
      set Flatlist = 'flatlist.dat';
      if ( -e $Flatlist ) then
         echo "! Removing old $Flatlist.";
         rm $Flatlist;
      endif
      touch $Flatlist;

      \rm flat_err.log
      touch flat_err.log

#  Set loop parameters and go through the list of files,
#  checking whether each file exists, process it,
#  and add its name to the flat-field frame list file.
      @ found = 0;
      foreach File ( $Files )

#        Locate raw file in data acquisition directory

         \cp -f $RawDir/$File $RedDir;
         \cp -f $RawDir/$File:r'.sdf' $RedDir;

#         chmod u+rwx $RedDir/$File ;
#         bunzip2 -f $RedDir/$File ;
         cd $RedDir ;

#      makendfs $File $CCD
#      \rm -f $File

#         set FitFile = $File:r ;

         set RawFile = $File:r ;
```

```
        echo \\n-----------------------------------------------------------
----
        echo Now working with $RawFile'.sdf' ;
#        fitsdin files=$FitFile out=$RawFile fmtcnv ;

        if ( -e $RawFile.sdf ) then
#           \rm $FitFile ;

#       Test for bad values
           istat $RawFile reset \\
             echo $ADAM_USER
             ls $ADAM_USER
           set string1 = `hdstrace $ADAM_USER/istat | grep STAT_TOTAL`

         set string2 = `hdstrace $ADAM_USER/istat | grep STAT_MAX`

         if ($string1[3] != "NaN") then

#       Subtract the bias frame from the input image.
             isub $RawFile $Biasframe junk1;

#       Use FIGARO ISTAT to determine the median level in the
#       selected overscan region.
             istat junk1 $ybimin $ybimax $xbimin $xbimax median;

#       Use FIGARO ICSUB to subtract the overscan level from the image.
#       This accesses the parameters written by the previous (ISTAT)
#       command and uses the median value found in the overscan region.
             icsub junk1 @$ADAMDIR/istat.STAT_MEDIAN junk2;

#       Set the name of the trimmed, debiassed image.
             set NewFile = f_$RawFile;

#       Trim off the unwanted parts of the frame, leaving only the
```

```
#         region containing the science data.
              isubset junk2 $Ymin $Ymax $Xmin $Xmax $NewFile;

#         Add the name of this file to the list of input frames for
#         FIGARO MEDSKY.
              echo $NewFile >> $Flatlist;

#         Increment the count of flat-field frames found.
              @ found++;

#         Delete intermediate files.
              \rm -f junk* $RawFile.sdf;

            else
              echo " $RawFile $string1[3] $string2[3] " >> flat_err.log
#           endif
          else
            echo "! Could not find $RawFile.";
          endif
        end

#  Display summary of preprocessing.
        echo " ! $found fields subtracted by master-bias, trimmed and ";
        echo " ! written to $Flatlist.";

#  Run FIGARO MEDSKY to generate the final frame and rotate it if needed.
        if ( $found != 0 ) then

echo
echo "Running MEDSKY"
          medsky $Flatlist scaled junk3;
echo

#       If your orders run with the dispersion direction in a roughly
#       vertical direction then uncomment last line of this comment.
```

```
#
#     Ideally, orders should run with wavelength increasing from left to
#     right across the frame - if an anticlockwise rotation is required
#     (rather than the default clockwise) then add the work "ANTI" to the
#     irot90 command.  Do not worry if you do not know which way the
#     wavelength scale goes.

#         irot90 IMAGE=junk3 OUTPUT=$Flatfield noanti;
#      irevx $Flatfield $Flatfield
#         \rm -f junk3.sdf

         irot90 junk3 junk4 anti;
         irevy junk4 $Flatfield
#         \mv -f junk3.sdf $Flatfield'.sdf'

      endif

#  Display informational message.
      echo "! Flat-field frame written to $Flatfield.";

#  Make balance frame (fibre data only).
      ixsmooth IMAGE=$Flatfield SIGMA=50 WIDTH=50 OUTPUT=smoothed;
      idiv smoothed $Flatfield $BalanceFr;
      echo "! Balance frame written to $BalanceFr.";
      \rm -f smoothed.sdf;


   endif


#  Reset any user command aliases.
      alias rm $rm_old;
      \rm -f f_*.sdf
```

```
#  End-of-file.
```

4.4 prepobjs

```
#!/bin/tcsh

#+
#  Name:
#     prepobjs

#  Purpose:
#     Script to prepare a set of object frames for use by ECHOMOP.

#  Language:
#     C shell script.

#  Description:
#     This script takes a list of "raw" CCD object frames and performs
#     the basic CCD data processing steps of bias subtraction and trimming
#     on the images.  Bias subtraction removes the zero-point bias
#     introduced by the camera electronics.  Trimming removes the pre-scan
#     and over-scan regions of the CCD image which contain no science data
#     and can confuse the algorithms in echelle data reduction software.
#
#     If required, the CCD frames can be rotated so that the dispersion
#     direction of the echelle orders runs horizontally as required by
#     ECHOMOP.
#
#     The script produces a series of files o_"file" where "file" is the
#     name of the source frame.

#  Usage:
#     This script can simply be invoked from the shell; in this case
#     the script will prompt for a list of the input object images.
#     Alternatively, the list of frames can be supplied on the
```

```
#     command line, for example:
#
#        % prepobjs run0800 run0801 run0802 run0856 run0857 run0858
#
#     If wildcarding facilities are available in your shell, you can use
#     them to simplify the command line, for example, the above would
#     become:
#
#        % prepobjs run080[012] run085[678]
#
#     in the C shell.  This wildcarding facility is available when the
#     script prompts for a list of input images.
#
#     In practice, invocation from your shell is unlikely to be a good
#     method of using this script as 8 environment variables defining
#     the region of the image to be kept and the region of the overscan
#     to be used for debiasing are required.  Use of these variables is
#     summarised below.

#  Notes:
#     1. If needed, the input parameters can be input at the command
#        line thus:
#
#           % nohup prepobjs filename [filename...] &
#
#        the "nohup" command will ensure that the script continues
#        to run even when you have logged off the system.  The "&" at
#        the end of the line will run the script in the background.
#
#     2. This script is designed to be used as part of an automated
#        echelle data reduction package.  If you intend to use it
#        for this purpose, you should not change the name of the output
#        frames from o_"file" where "file" is an input frame.  See the
#        comments in the script for changes which can be made if it is
```

```
#       to be used stand-alone.
#
#    3. This script will work with FIGARO v5.0-0 or later.
#
#    4. When this script is invoked, 8 environment variables defining
#       the overscan region to be used for debiasing, and the region
#       of the images containing science data must be defined.
#       These environment variables are used:
#
#        $xbimin
#          X-start of overscan region to use for bias subtract.
#        $xbimax
#          X-end of overscan region to use for bias subtract.
#        $ybimin
#          Y-start of overscan region to use for bias subtract.
#        $ybimax
#          Y-end of overscan region to use for bias subtract.
#        $xtrmin
#          X-start of region of image to be retained.
#        $xtrmax
#          X-end of region of image to be retained.
#        $ytrmin
#          Y-start of region of image to be retained.
#        $ytrmax
#          Y-end of region of image to be retained.
#
#    5. A file "biasfram" containing a bias frame prepared by the
#       script "prepbias" should exist in the working directory.
#       You can alter the name of this file, see comments in the
#       script.
#
#    6. The input frames are NOT rotated by this script.  You may
#       have images in which the orders run roughly vertically,
#       in which case you should uncomment the line using FIGARO
#       IROT90 as indicated in the script. Approximately horizontal
```

```
#        orders are required by ECHOMOP.
#
#     7. This script is designed to be called by a master reduction
#        script.  See the example scripts "preprun1" and "preprun2"
#        for details.

#  Authors:
#     ACC: Andrew Collier Cameron (St. Andrews)
#     MJC: Martin Clayton (Starlink)
#     CSM: Calum McCune (Open University)

#  History:
#     ??-???-???? (ACC):
#       Original Version.
#     06-NOV-1995 (MJC):
#       Minor tidying up for Cookbook Version.
#     JUN-2015 (CSM):
#     Minor edits made so that script can run within pipeline and
#     with SOPHIE archival data

#-

#  Do Starlink login and set up for Starlink applications used.
#  You can comment out these lines if you already have these set up AND
#  you invoke the script by "source"ing it, for example:
#
#     % source prepobjs run080[012] run085[678]
#
      #source /star/etc/cshrc;
      #source /star/etc/login;
      star32
      alias echo 'echo >/dev/null';
      setenv ADAM_USER prepobjs_adam
      \rm -rf $ADAM_USER
      mkdir $ADAM_USER
```

```
      figaro;
      kappa;
      unalias echo;

#  Force standard UNIX commands to "normal" behaviour.
#  This is to remove any special alias for the remove command.
      set rm_old = `alias rm`;
      unalias rm;

#  This line should only be modified if you wish to run several jobs
#  simultaneously, or you keep your Starlink parameter files in a
#  non-default directory.  If you do not understand this then leave the
#  line as it is!
      set ADAMDIR = $ADAM_USER;

#  Get the list of input object frames, either from the command line
#  or by prompting.
      if ( "$#argv" == 0 ) then
         echo '';
         echo -n '? Object Frame(s) > ';
         set Files = ( `echo $< ` );

      else
         set Files = ( $* );
      endif

#  Display information confirming the selected bias-level calculation
#  region.
      echo " Bias statistics will be calculated over X-range "\
 "$xbimin to $xbimax ";
      echo "                                    and over Y-range "\
 "$ybimin to $ybimax ";

#  Display information confirming the selected region of the image to be
#  retained; i.e., the CCD area containing the science data.
      echo " Frames will be trimmed over X-range $xtrmin to $xtrmax ";
```

```
      echo "                    and over Y-range $ytrmin to $ytrmax ";

#  Set script-internal variables to describe the trimming region.
      set Xmin = $xtrmin;
      set Xmax = $xtrmax;
      set Ymin = $ytrmin;
      set Ymax = $ytrmax;

#  Set the name of the input median bias frame.
#  You can change this line if you are using the script on its own.
      set Biasframe = $Bias;

#  Set loop parameters and go through the input file list,
#  checking whether the files exist, and processing them into "o_" files.
      @ found = 0;
      foreach File ( $Files )

#        Locate raw file in data acquisition directory

         \cp -f $RawDir/$File $RedDir ;
         \cp -f $RawDir/$File:r'.sdf' $RedDir ;
#         chmod u+rwx $RedDir/$File ;
#         bunzip2 -f $RedDir/$File ;
         cd $RedDir ;
#
#      makendfs $File $CCD
#      \rm -f $File

#         set FitFile = $File:r ;

         set RawFile = $File:r ;

         echo \\n-----------------------------------------------------------
----
         echo Now working with $RawFile'.sdf' ;
```

```
#          fitsdin files=$FitFile out=$RawFile fmtcnv ;

           if ( -e $RawFile.sdf ) then
#         Subtract the bias frame from the input image.
             isub $RawFile $Biasframe junk1;

#         Use FIGARO ISTAT to determine the median level in the
#         selected overscan region and subtract it.

    istat junk1 xstart=$xbimin xend=$xbimax ystart=$ybimin yend=$ybimax
median;
           set med1 = `hdstrace $ADAM_USER/istat.STAT_MEDIAN`
           set medval = $med1[4]

           icsub junk1 $medval junk2

#         Set the name of the trimmed, debiassed image.
             set NewFile = o_$RawFile;

#         Trim off the unwanted parts of the frame, leaving only the
#         region containing the science data.

             isubset junk2 ystart=$Ymin yend=$Ymax xstart=$Xmin xend=$Xmax
output=junk3;

             irot90 junk3 junk4 anti;
             irevy junk4 $NewFile
#           \mv -f junk3.sdf $NewFile'.sdf'

#         Increment the count of flat-field frames found.
             @ found++;

#         Delete intermediate files.
             rm junk*
            rm $RawFile.sdf ;
```

```
         else
            echo "! Could not find $RawFile";
         endif
      end

#  Display informational message.
      echo "! $found object frames subtracted by master-bias, trimmed and
rotated.";

#  Reset any user command aliases.
      alias rm $rm_old;



#  End-of-file.
```

4.5 preparcs

```
#!/bin/tcsh

#+
#  Name:
#     preparcs

#  Purpose:
#     Script to prepare a set of arc frames for use by ECHOMOP.

#  Language:
#     C shell script.

#  Description:
#     This script takes a list of "raw" CCD object frames and performs
#     the basic CCD data processing steps of bias subtraction and trimming
#     on the images.  Bias subtraction removes the zero-point bias
#     introduced by the camera electronics.  Trimming removes the pre-scan
#     and over-scan regions of the CCD image which contain no science data
```

```
#     and can confuse the algorithms in echelle data reduction software.
#
#     The script produces a series of files a_"file" where "file" is the
#     name of the source frame.

#  Usage:
#     This script can simply be invoked from the shell; in this case
#     the script will prompt for a list of the input object images.
#     Alternatively, the list of frames can be supplied on the
#     command line, for example:
#
#        % preparcs run0800 run0801 run0802 run0856 run0857 run0858
#
#     If wildcarding facilities are available in your shell, you can use
#     them to simplify the command line, for example, the above would
#     become:
#
#        % preparcs run080[012] run085[678]
#
#     in the C shell.  This wildcarding facility is available when the
#     script prompts for a list of input images.
#
#     In practice, invocation from your shell is unlikely to be a good
#     method of using this script as 8 environment variables defining
#     the region of the image to be kept and the region of the overscan
#     to be used for debiasing are required.  Use of these variables is
#     summarised below.

#  Notes:
#     1. If needed, the input parameters can be input at the command
#        line thus:
#
#           % nohup preparcs filename [filename...] &
#
#        the "nohup" command will ensure that the script continues
```

```
#       to run even when you have logged off the system.  The "&" at
#       the end of the line will run the script in the background.
#
#    2. This script is designed to be used as part of an automated
#       echelle data reduction package.  If you intend to use it
#       for this purpose, you should not change the name of the output
#       frames from a_"file" where "file" is an input frame.  See the
#       comments in the script for changes which can be made if it is
#       to be used stand-alone.
#
#    3. This script will work with FIGARO v5.0-0 or later.
#
#    4. When this script is invoked, 8 environment variables defining
#       the overscan region to be used for debiasing, and the region
#       of the images containing science data must be defined.
#       These environment variables are used:
#
#        $xbimin
#          X-start of overscan region to use for bias subtract.
#        $xbimax
#          X-end of overscan region to use for bias subtract.
#        $ybimin
#          Y-start of overscan region to use for bias subtract.
#        $ybimax
#          Y-end of overscan region to use for bias subtract.
#        $xtrmin
#          X-start of region of image to be retained.
#        $xtrmax
#          X-end of region of image to be retained.
#        $ytrmin
#          Y-start of region of image to be retained.
#        $ytrmax
#          Y-end of region of image to be retained.
#
```

```
#     5. A file "biasfram" containing a bias frame prepared by the
#        script "prepbias" should exist in the working directory.
#        You can alter the name of this file, see comments in the
#        script.
#
#     6. This script is designed to be called by a master reduction
#        script.  See the example scripts "preprun1" and "preprun2"
#        for details.

#  Authors:
#     ACC: Andrew Collier Cameron (St. Andrews)
#     MJC: Martin Clayton (Starlink)
#     CSM: Calum McCune (Open University)

#  History:
#     ??-???-???? (ACC):
#     Original Version.
#     JUN-2015 (CSM):
#     Minor edits made so that script can run within pipeline and
#     with SOPHIE archival data
#


      #source /star/etc/cshrc;
      #source /star/etc/login;
      star32
      alias echo 'echo >/dev/null';
      setenv ADAM_USER preparcs_adam
      \rm -rf $ADAM_USER
      mkdir $ADAM_USER
      figaro;
      kappa;
      unalias echo;


#  Force standard UNIX commands to "normal" behaviour.
#  This is to remove any special alias for the remove command.
      set rm_old = `alias rm`;
      unalias rm;
```

```
#  This line should only be modified if you wish to run several jobs
#  simultaneously, or you keep your Starlink parameter files in a
#  non-default directory.  If you do not understand this then leave the
#  line as it is!
      set ADAMDIR = $ADAM_USER;

#  Get the list of input ThAr frames, either from the command line
#  or by prompting.
      if ( "$#argv" == 0 ) then
         echo '';
         echo -n '? Arc Frame(s) > ';
         set Files = ( `echo $< ` );

      else
         set Files = ( $* );
      endif

#  Display information confirming the selected bias-level calculation
#  region.
      echo " Bias statistics will be calculated over X-range "\
 "$xbimin to $xbimax ";
      echo "                                  and over Y-range "\
 "$ybimin to $ybimax ";

#  Display information confirming the selected region of the image to be
#  retained; i.e., the CCD area containing the science data.
      echo " Frames will be trimmed over X-range $xtrmin to $xtrmax ";
      echo "                       and over Y-range $ytrmin to $ytrmax ";

#  Set script-internal variables to describe the trimming region.
      set Xmin = $xtrmin;
      set Xmax = $xtrmax;
      set Ymin = $ytrmin;
      set Ymax = $ytrmax;
```

```
#  Set the name of the input median bias frame.
#  You can change this line if you are using the script on its own.
      set Biasframe = $Bias;

#  Set loop parameters and go through the input file list,
#  checking whether the files exist, and processing them into "o_" files.
      @ found = 0;
      foreach File ( $Files )

#        Locate raw file in data acquisition directory

         \cp -f $RawDir/$File $RedDir ;
         \cp -f $RawDir/$File:r'.sdf' $RedDir ;

#         chmod u+rwx $RedDir/$File ;
#         bunzip2 -f $RedDir/$File ;
         cd $RedDir ;
#
#      makendfs $File $CCD
#      \rm -f $File

#         set FitFile = $File:r ;

         set RawFile = $File:r ;

echo \\n------------------------------------------------------------
         echo Now working with $RawFile'.sdf' ;
#         fitsdin files=$FitFile out=$RawFile fmtcnv ;

          if ( -e $RawFile.sdf ) then

#        Subtract the bias frame from the input image.
            isub $RawFile $Biasframe junk1;
```

```
#         Use FIGARO ISTAT to determine the median level in the
#         selected overscan region and subtract it.

            istat junk1 xstart=$xbimin xend=$xbimax ystart=$ybimin
yend=$ybimax median;
           set med1 = `hdstrace $ADAM_USER/istat.STAT_MEDIAN`
           set medval = $med1[4]

           icsub junk1 $medval junk2

#         Set the name of the trimmed, debiassed image.
            set NewFile = a_$RawFile;

#         Trim off the unwanted parts of the frame, leaving only the
#         region containing the science data.

            isubset junk2 ystart=$Ymin yend=$Ymax xstart=$Xmin xend=$Xmax
output=junk3;

            irot90 junk3 junk4 anti;
            irevy junk4 $NewFile
#           \mv -f junk3.sdf $NewFile'.sdf'

#         Increment the count of flat-field frames found.
            @ found++;

#         Delete intermediate files.
            rm junk*
           rm $RawFile.sdf ;

         else
            echo "! Could not find $RawFile";
         endif
      end
```

```
#  Display informational message.

      echo "! $found ThAr frames subtracted by master-bias, trimmed and
rotated.";


#  Reset any user command aliases.

      alias rm $rm_old;



#  End-of-file.
```

Appendix 5: Tweek_pipeline Script (cshell)

```
#!/bin/tcsh

#
#  Name:
#     tweek_pipeline
#
#  Purpose:
#     Script to automate the running of the 'preprun' scripts for all raw
#     nights data folders in the current working directory.
#
#  Operation:
#     Script identifies all target folders in the current working
#     directory as those with the names T0...SOPHIE..., where the prefix
#     'T0' is the 'KEYWORD' (declared within this script can be editted
#     by user as appropriate) used to identify the nights data folders.
#     This nomenclature should be setup by the user before execution. The
#     names of the nights data folders are then stored as a list which is
#     used by a for loop uses to run through the following processes for
#     each:
#           - changes directory to T0XX.../raw/'date'
#           - stores the date information in the directory title as a
#             string in a form that can be used by upcomig scripts.
#             i.e. 2008-02-20 > 20080220
#           - copies all necessary cshell scripts into 'date directory'
#           - runs script 'makelog' to create the log file used to
#             determine the observation types of each data file.
#           - runs script 'make_ndfs' to rename files for Starlink
#             format and creates copies in ndf format.
#           - runs script 'preprun_master' which calls all other
#             scripts to perform all pre-ECHOMOP reductions
#
#  Usage:
#     To run this script it must exist in the directory containing the
```

```
#     raw nights data folders accompanied by the folder 'cshell_scripts'
#     which contains the 'preprun' scripts. All of the raw nights data
#     folder must be renamed with the prefix 'T0..' corresponding to the
#     'KEYWORD'.
#
# Author:
#    Calum McCune, 2015 (Open University)



### Defining Base directory & creating directory list ###
      setenv Dir `echo $cwd`  # sets Dir to store the current base
                              # directory which contains all nights data
                              # directories.

      echo "Base Directory: $Dir"   # status check

      ############ KEYWORD ############
      # In order to correctly identify the folders containing the sophie
      # raw nights data, enter a key word that is common to THE START of
      # each directory name.
      #set KEYWORD = T0       # change to be whatever is common to all
                              # data folders
      set KEYWORD = HR0       # alternate created to deal with new HR
                              # mode nights
      echo $KEYWORD           # status check
      ###################################

      set dirs = `find $Dir -name "*$KEYWORD*" -type d -print` # creates
                              # list of directory names which contain
                              # night data.

      set dirs = `echo $dirs | fmt -1 | sort -n` # alphabetises list for
                              # ease of following execution
#########################################################
```

```
      ### Defining 'script directory' location ###
      setenv script_dir $Dir/cshell_scripts # sets script_dir to hold the
                        # location string of the directory
                        # containing the cshell scripts to be
                        # copied into each data directory.
      echo "Script Directory: " $script_dir:t # status check
      ##############################################




### FOR LOOP to run for all night data direcotries ###
      foreach dir ($dirs)     # the string 'dir' stores the location path
                        # of each directory listed in 'dirs'
                        # NOTE: $dirs[2-5] can be used to loop
                        # through the just the directories in the
                        # range 2 - 5 from the list 'dirs'

        echo \\n\\n
"*********************************************************"
        echo \\n \\t"NEW TARGET"
        echo \\n "*********************************************************"
        echo \\n\\n
        echo \\n"Working with data folder: "$dir:t # status check, :t
                        # removes file pathway


        set start_time = `date +'%A %d/%m/%Y %T.%N'`  # code used to time
                        # execution
        set start_JD = `date +'%s.%N'`


        ### Finding 'CURRENTDATE' ###
        # Following code creates the string 'CURRENTDATE which holds the
        # observation date information required as the input arguement to
        # the script 'preprun_master' and used by all subsequent scripts.
```

```
set Date = `find $dir/raw -maxdepth 1 -mindepth 1 -type d -printf
'%f\n'`
                # sets 'Date' to hold the date information
                # as read from the data directory name
echo "Date: $Date"    # status check

setenv CURRENTDATE `echo $Date | sed 's/\-//g'` # formats date to
                # be used for later scripts by removing the
                # symbol '-'
echo "CURRENTDATE: $CURRENTDATE" # status check

cd $dir/raw/$Date     # changes working directory to the data
                # directory


###### WITHIN DIRECTORY ######
cp $script_dir/* .    # copies all scripts from script directory
                # into current directory
./makelog       # creates log file of all FITS file contents
./make_ndfs           # executes script to convert all FITS files
                # to the format and naming syntax native to
                # starlink
./preprun_master      # master script that invokes all subsequent
                # 'preprun' scripts to prepare the frames
                # used by ECHOMOP
cd $Dir         # returns to Base directory ready to repeat
                # for next nights directory
##############################

### Execution Timing ###
echo \\n"Start time : $start_time"
set end_time = `date +'%A %d/%m/%Y %T.%N'`
echo "End time : $end_time"

set end_JD = `date +'%s.%N'`
set exe_time = `echo "$end_JD - $start_JD"|bc`
```

```
        set DAYS = `echo "scale=0; $exe_time/(24*3600)"|bc`
        set subtime = `echo "scale=0; $exe_time - ($DAYS*24*3600)"|bc`
        set HOURS = `echo "scale=0; $subtime/3600"|bc`
        set subtime = `echo "scale=0; $subtime - ($HOURS*3600)"|bc`
        set MINS = `echo "scale=0; $subtime/60"|bc`
        set subtime = `echo "scale=0; $subtime - ($MINS*60)"|bc`
        set SECS = $subtime

        echo "Execution time : $DAYS days $HOURS hours $MINS minutes $SECS
secs"\\n\\n
        ##########################

    end
```

Appendix 6: ECHOMOP Walkthrough

Here I give a walkthrough of the inputs to run the ECHOMOP Échelle reduction software from the Linux Terminal on a given observation night's data frames. The frames mentioned below are all given as outputs from the CCD frame preparation scripts detailed in appendices 3 & 4.

Open the terminal in the Directory of the reduced Échelle spectra frames

From here run ECHOMOP by entering the commands:

> star32

> figaro

> kappa

> echomop

ECHOMOP is now setup and ready to be run.

Run the guided ECHMOP pipeline using the command:

> echmenu

Enter name of the Reduction Database file (either that already exists, or, to be created):

ECH_RDCTN > <name of reduction database file> ← stores all information for the reduction

Run Option 1

Option number > 1

TRACIM > <frame for order tracing> ← use flt field frame for clearest orders

INPTIM > <object file>

ARC > None

A window will now open allow user to check the orders that have been identified. From here delete all false peaks/non-object peaks using click + D, once finished use 'e' to exit.

Run Option 2

Option number > 2

TRCFIT > 'POLY' ← option for order tracing

TRACE_MODE > 'C'

TRC_NPOLY > <no. of coefficients to be used> '7'

Run Option 3

Option number > 3

TRC_INTERACT > TRUE

A window for clipping the order trace fit will now open. Use click, then 'C' to clip the deviation about the mean, and 'e' to exit to next order, alternatively 'g' for go to automatically run for all orders.

Run Option 4

Option number > 4

PFL_INTERACT > TRUE

SLITIM > <fltfield frame>

Dekker determination window will open, allowing user to define the limits of the Dekker.

use 'L' & 'U' keys & positioning cursor to set the lower and upper limits.

Use '+' and '-' to cycle through orders

'e' to exit when done

Next the object/sky determination window will open. Allowing user to define the limits of the object and sky background pixels.

same control options as above

'e' to exit when done

Configure Option 5

Option number > -5

check 16. = TRUE, if not enter 16, then type True. 0 to exit when done.

Run Option 5

Option number > 5

FFELD > <balance frame>

FLTFIT > 'MEAN'

Choose Background subtraction option:

To run Option 6:

Option number > 6

SKYFIT > None

PHOTON_TO_ADU > <Gain> '1' ← use 1 unless a different value of gain is needed

READOUT_NOISE > <Noise> '0' check fits header. Dame for Noise = 0

Alternatively, to run Option 22

Option number > 22

Give same inputs as Option 6

Or if using external background subtraction, skip to Option 8.

Run Option 7

Option number > 7

Run Option 8

Option number > 8

EXTRACT_MODE > O

Finally save the output using Option 14

Option number > 14

RESULT_FORMAT > 'NDF'

RESULT_TYPE > 'EXOBJ'

ECH_RDUCD > <output file>

= DONE = 'q' to quit

Appendix 7: SKYBM Background Subtraction Code (Python)

7.1 SKYBM_functions_v1707.py

```
# Name:
#   SKYBM_functions_v1707
#
# Purpose:
#    Script serves as a library of functions I devloped for specific tasks in
#    the 'SKYBM' process scripts.
#
# Operation:
#    Providing that this script exists in the same working directory, other
#    scripts import it as a library and invoke the functions here in.
#
# Author
#    Calum McCune, 2015 (Open University)


######## LIBRARIES #################
import numpy as np
import pyfits
import sys
import scipy.interpolate as inter
###################################



###### FUNCTIONS ##################

def file_input():
    ''' () -> string

    Overveiw:
        Function prompts the user to choose the fits file to open. Returns that
        choice as a string.

    Description:
```

```
    The user is provided with a selection of preexisting fits files to open,
    in addition the choice to enter any filename is given. The code includes
    fail safes for incorrect entrys. If an input other than the options
    provided is given the function will select the file fitstestfile.fits as
    default. If the file extension .fits is not included in the name of an
    entered file then the extension will be added.



'''
print 'Which FITS file would you like to open?:' # User presented choice of
print '\t1. fitstestfile.fits'  # three fitsfile options which were used
print '\t2. o_fitstest2.fits'   # during the development and implementation
print '\t3. o_fitstest3.fits'   # of this program. In addition the choice
print '\t0. Enter other filename' # to enter any filename is given.

# if statements to respond to user choices and to assign correct filename.
selection = raw_input('Please make a selection: ')
if not selection:
    print 'No selection made opening fitstestfile.fits as default'
    fitsfile = 'fitstestfile.fits'
elif (selection == '1'):
    print '\n\tOption 1 chosen, opening fitstestfile.fits'
    fitsfile = 'fitstestfile.fits'
elif (selection == '2'):
    print '\n\tOption 2 chosen, opening o_fitstest_2.fits'
    fitsfile = 'o_fitstest_2.fits'
elif (selection == '3'):
    print '\n\tOption 3 chosen, opening o_fitstest_3.fits'
    fitsfile = 'o_fitstest_3.fits'
elif (selection == '0'): # in the case of option 0 the user can input their
                # desired filename. In this situation the user could enter
                # the correct filename, the filename w/o the file type i.e.
                # .fits, or not enter anything at all. The code contained in
                # this if statement deals with each of these three
                # eventualities.
    filename = raw_input('\nPlease enter the fits file name: ')
    if not filename:
        print '\nNo filename was entered.'
```

```
            print '\nOpening fitstestfile.fits as default'
            fitsfile = 'fitstestfile.fits'
        else:
            if (filename[-5:] == '.fits'):
                fitsfile = filename
                print filename, 'opened!'
            elif (filename[-5:] != '.fits'):
                fitsfile = filename + '.fits'
                print fitsfile, 'opened!'
    else:           # in the case of an invalid selection the program will
                    # default to use the file fitstestfile.fits.
        sys.stdout.write('Invalid selection!')
        print '\nOpening fitstestfile.fits as default'
        fitsfile = 'fitstestfile.fits'
    return fitsfile



def open_FITS(FITS_file):
    ''' (string) -> array

    Overveiw:
        Function opens a given fits file and extracts the image data to an
        array.

    Arguments:
        FITS_file: string containing the fits file name obtained from the
            function file_input()

    Description:
        Opens the allocated fits file and extracts the science image data from
        the relevant fits extension to an array. Also creates an arbitrarily
        sized subarray which is more managable for code testing. The user is
        then given the choice of implementing either of these two arrays. The
        resulting choice being the return array.


    '''
    hdu_list = pyfits.open(FITS_file)
```

```
    print '\n', FITS_file,' open\n'
    hdu = hdu_list[0] # NOTE: FITS files can have multiple extensions,
                      # 0 references the first.
    image = hdu.data # array containing the image data from extension [0]

    raw_input('\nPress any key to continue...\n\n')
    return image



def mask_open():
    ''' (string) -> array

    Overveiw:
        Function opens the given mask containing FITS file and extracts the
        image data to an array.

    Arguments:
        None: User prompted for input within function

    Description:
        Prompts the user to enter the name of the mask FITS file to be used.
        If the file extension .fits is not included in the name of an entered
        file then the extension will be added. Attempts to open the allocated
        FITS file, if successful extracting the science image data from
        the relevant fits extension to an array. If attempt is unsuccessful
        then the file does not exist in the directory and excetution is
        terminated.


    '''

    print 'Please enter the name of the fits file containing the MASK you would you
like to use' # User presented choice of

    filename = raw_input('MASK file name: ')

    if (filename[-5:] == '.fits'):
        mask_file = filename
```

```
    elif (filename[-5:] != '.fits'):
        mask_file = filename + '.fits'

    try:
        hdu_list = pyfits.open(mask_file)
        print '\n', mask_file,' open\n'
        hdu = hdu_list[0]
        MASK = hdu.data
        raw_input('\nPress any key to continue...\n\n')
        print 'Running, please wait...\n\n'
        return MASK
    except:
        print mask_file, 'does not exist!'



def spline_med(cut,N):
    '''(array,int) -> (array)

    Overveiw:
        Function produces a Univariate spline fit to a median binned array

    Arguments:
        cut: array to fit spline to
        N: integer number of sections 'cut' array divided into for median
            binning

    Description:
        This function breaks the data in the input array 'cut' into 'N' equal
        sections. The central pixel locations of each of these sections is then
        stored in an array as X values. The median values of each of these
        sections is calculated and store in an array as Y values. These
        'median-binned' X and Y values are then used to generate the Univariate
        spline function which can then produce a line fitting this data for any
        size of array given as input. An array of linearly increase values the
        size of 'cut' is made and used as input for the Univariate spline
        function to generate a spline array of equal size to 'cut'.
        This array is then returned as the output.
```

```
    '''
    sl = np.linspace(0,len(cut),N) # creates a "straight line" array of N
                    # evenly spaced values between zero and len(cut).

    X = ((sl[1:]-sl[:-1])//2) + sl[:-1] # array X holds the values halfway
                    # between each of the elements in the array 'sl'. These
                    # values represent the central pixel locations of each
                    # of the median bin regions.
    Y = np.array([]) # empty array created to hold Y values

    for m in range(0,N-1): # for loop to fill the Y array with the local median
                    # values of each of the N median bin regions.
        M = cut[sl[m]:sl[m+1]] # sub array M holds each of the N regions
                    # sequentially.
        sl_med = np.median(M) # takes the median value of each region
        Y = np.append(Y,sl_med) # stores each sequential median value in the Y
                    # array.

    X_new = np.linspace(0,len(cut)-1,len(cut)) # new array to hold X axis
                    # values for the new polynomial line. This array is a
                    # straight line of evenly spaced values of equal size to
                    # the cut array.
    S = s_est(Y)    # function s_est() generates the best estimate of the
                    # 's' value, needed for the spline function, for the 'cut'
    Y_new = inter.UnivariateSpline(X,Y,s=S) # produces the spline function for
                    # the given X, Y and S data.
    return Y_new(X_new)



def inter_points(cut,mask_cut):
    '''(array,array) -> ([array,array])

    Overveiw:
        Function produces two arrays containing the central pixel location and
        local median values of all regions in the input array 'cut' not masked
        by the mask array 'mask_cut'. Arrays returned as outputs [0] and [1]
        respectively
```

```
Arguments:
    cut: cross-section array of echelle image
    mask_cut: corresponding cross-section array of mask

Description:
    This function uses the array 'mask_cut' to identify the masked and
    un-masked regions in the array 'cut'. Where the pixels in 'mask_cut'
    have the value 0, they are considered un-masked and where they have the
    value 100, they are considered masked. A for loop reads through every
    pixel in both arrays. In regions that are un-masked the pixel values
    are stored in the array 'bunch'. When this un-masked region ends the
    median and the length of 'bunch' is taken, the median is stored in
    'med_data' and the length is halved and subtracted from the current
    pixel location 'i' to give the central pixel location of this region,
    which is then stored in 'mid_loc'. At the end of every un-masked region
    'bunch' is then reset ready to repeat the  process for the next
    un-maksed region until the end of the 'cut'. The two arrays 'mid_loc'
    and 'med_data' are then returned as outputs [0] and [1] respectively.

'''
bunch = np.array([])     # initiailizes empty array to hold pixel data from
                         # the unmasked region
med_data = np.array([]) # initiailizes empty array to hold the inter-mask
                         # median data
mid_loc = np.array([])  # initiailizes empty array to hold the central
                         # unmasked region locations

## for loop to generate interpeak central median points MASK
for i in range(0,len(cut)):         # loop performs operation for each
                                    # pixels in the cut
    if (mask_cut[i] == 0)&(i != (len(cut)-1)): # if condition met when mask
                         # pixel is equal to zero, i.e. unmasked, and not
                         # the final pixel
        bunch =np.append(bunch,cut[i]) # apend the cut pixel to bunch array
    elif ((mask_cut[i] == 100)|(i == (len(cut)-1)))&(bunch.size != 0):
                         # elif cut pixel is masked and bunch array is not
                         # empty
        d = (bunch.size+1)/2 # divides the size of bunch by 2 to get the
                         # half width of the unmasked region
```

```
            x = i - d       # x = pixel location of mid point of unmasked
                            # region
            mid_loc = np.append(mid_loc,x) # central pixel location appended to
                            # mid_loc array.
            med_data = np.append(med_data, np.median(bunch)) # median value of
                            # current bunch appended to med_data array.
            bunch = np.array([]) # redefines bunch as an empty array

    return mid_loc,med_data # returns the two arrays holding the mid-points of
                            # the unmaksed regions, and the median values of
                            # each unmasked region respectively


def s_est(cut):
    '''(array) -> (int)

    Overveiw:
        Function to generate the best estimate of 's', for the current array.
        This value is smoothing needed for the univariate spline function.

    Arguments:
        cut: cross-section array of echelle image

    Description:
        This function uses the formula given in the documentation for the
        inter.UnivariateSpline() function to estimate the best value of the
        smoothing factor 's' for the given array. This value is required by the
        inter.UnivariateSpline() function to create the spline function that
        renders the spline for a given size.

    '''
    m = cut.size        # m no. of data points
    s_min = (m - np.sqrt(2*m))*np.std(cut)**2
    s_max = (m + np.sqrt(2*m))*np.std(cut)**2
    s_est = (s_max+s_min)/2

    return s_est
```

```
def array_size(image):
    '''(array) -> ([array,int,int,int,int])

    Overveiw:
       Function for re-sizing the image array

    Arguments:
        image: the echelle image data array

    Description:
        In the case where a smaller region of the CCD is wanted to be used,
        this function allows for the image array to be resized to the users
        given dimensions. The user is prompted for the X_min, X_max, Y_min and
        Y_max start and end points and the image array is trimmed accordingly.
        These chosen dimensions are also returned by the function as outputs so
        that all code dependent on this arrays shape can be changed accordingly.

    '''

    choice = raw_input('Use full array or user defined region? (full or sub): ')

    if (choice == 'full'):
        print '\nFull image array used'
        array = image
        X_min = 0
        X_max = image.shape[0]
        Y_min = 0
        Y_max = image.shape[1]
    elif (choice == 'sub'):
        print ("Array shape,\tx: %d pixels\n\t\ty: %d pixels"
%(image.shape[0],image.shape[1]))
        X_min = input('x min: ')
        X_max = input('x max: ')
        Y_min = input('y min: ')
        Y_max = input('y max: ')
        sub = image[X_min:X_max,Y_min:Y_max]
        array = sub
        print ('\nSub-array of image created')
```

```
    else:
        sys.stdout.write('Invalid selection!')

    return array, X_min, X_max, Y_min, Y_max
```

7.2 thresh_find_v1707.py

```
# Name:
#     thresh_find_v1707
#
# Purpose:
#    Script to find the 'threshold counts value' used to distinguish
between
#    the background noise and echelle data.
#
# Operation:
#    The script prompts the user to enter the name of the echelle spectra FITS
#    file to be worked on. The image data from this file is then extracted and
#    stored as an array. The user is then asked how many cuts they would like
#    to check, this is the number of evenly spaced vertical (perdenicular to
#    echelle orders) cross sections they would like to display so they can
#    manually check the threshold value is appropriate across the whole image.
#    The user is then prompted for an initial guess counts threshold value.
#    For each of the given number of 'cuts' a 'threshold line' (threshline)
#    will be created by generating a spline fit to 20 median binned data points
#    and increasing this spline by the given threshold counts value. This line
#    will be used by the mask making procedure to define data regions on the
#    image to be masked for the background modelling. Anything above this line
#    will be masked. Plots of each of the evenly spaced cuts are produced
#    showing the column data in grey and the threshold line in red for contrast.
#    The user can then veiw these plots and judge the appropriatenes of value
#    chosen and decide whether to use a greater or smaller value.
#    This script should be re-run until an appropriate threshold counts value
#    has been found.
#
#  Author:
#    Calum McCune, 2015 (Open University)
```

```
### SETUP ###
######## LIBRARIES ################
import numpy as np                # library numpy used for array manipulation
import SKYBM_functions_v1707 as sm  # library containing functions I
                                  # created specifically for use in the
                                  # 'SKYBM' pipeline
import matplotlib.pyplot as plt  # used for plotting
###################################

######## Constants ################
N = 20                            # number of median biined data points to
                                  # be used by the function 'sm.spline_med()'
                                  # to generate the initital median spline
                                  # fit to the data
###################################

######## data setup ###############
fitsfile = sm.file_input()        # User interactive function for selecting
                                  # echelle spectra FITS file
array = sm.open_FITS(fitsfile)    # function extracts image data from the
                                  # given FITS file to the array 'array'
###################################



def thresh_find(array):           # main function of the script
    cut_no = input('How many cuts would you like to check?: ') # prompts user
                                  # for desired number of cuts
    thresh_val = input('Enter threshold counts value: ') # prompts user for
                                  # threshold counts value
    for cut in range(1,cut_no+1):  # for loop performs all operations for each
                                  # of the chosen number of cuts

        # CODE for deciding the evenly spaced column numbers to be used given
        # the chosen number of cuts.
        sects = cut_no+1
        sect_size = array.shape[1]/float(sects)
```

```
        cut_loc = round(sect_size*cut)
        CUT = array[:,cut_loc]


        S_MED = sm.spline_med(CUT,N)    # function sm.spline_med() creates
                                        # the spline 'S_MED'. This spline has been
                                        # created by median binnig the data into 'N'
                                        # evenly spaced points, to which a
                                        # univariate spline is fitted
        threshline = np.zeros(len(CUT)) # 1d array of zeros of the same length
                                        # as the cut initialised to hold the
                                        # 'threshold line'
        threshline[:] = S_MED[:] + thresh_val # 'threshold line' created by
                                        # increasing all points on 'S_MED' by the
                                        # given counts threshold value

        # plotting data
        fig = plt.figure()
        plt.plot(CUT,'grey')            # column data plotted in grey
        plt.plot(threshline,'r')        # threshold line plotted in red
        title = 'Array column: '+`cut_loc`+'  Threshold value: '+`thresh_val`
        fig.suptitle(title,fontsize=16)
    plt.show()



thresh_find(array)                      # main function run for the image data
                                        # array extracted from the given FITS file
```

7.3 make_mask_vM1707.py

```
# Name:
#   make_mask_vM1707
#
# Purpose:
#    Script to produce a mask used to mask the echelle data regions for the
#    subsequent background modelling and subtracting procedure. The mask created
```

```
#    is then output as a seperate FITS file.
#
# Operation:
#    The script prompts the user to enter the name of the echelle spectra FITS
#    file to be worked on. The image data from this file is then extracted and
#    stored as an array. The user is then prompted for the counts threshold
#    value found using the prerequisite script 'thresh_find_v1707'. Next the
#    user is prompted to enter a pixel margin value which will be used to
#    increase the widths of the masked regions ensuring peak coverage. Using
#    this value a threshold line is generated and used to distinguish the
#    background noise from the data regions for each column in the image.
#    An initial empty array of zeros the same size as the image array is created
#    to hold the mask. All image data pixels exceeding the local threshold line
#    are considered as data region and the corresponding pixels in the mask
#    array are set to the value 100, with all background region pixels remaining
#    zero. The completed mask array is then output to a new FITS file titled
#    <input filename>_MASK.fits.
#    Additionally if a mask with this name already exists in the working
#    directory then this new file will be named with the suffix [1], or [2] etc.
#    if multiple versions exist.
#
#   Author:
#       Calum McCune, 2015 (Open University)



### SETUP ###
######## LIBRARIES #################
import numpy as np                # library numpy used for array manipulation
import pyfits                      # library pyfits used for working with FITS
                                   # files manipulation
import SKYBM_functions_v1707 as sm   # library containing functions I
                                   # created specifically for use in the
                                   # 'SKYBM' pipeline
import os.path                    # for checking files in directory
import time                        # used for execution timer
import matplotlib.pyplot as plt  # used for plotting
####################################
```

```
######## Constants ################
N = 20                              # number of median biined data points to
                                    # be used by the function 'sm.spline_med()'
                                    # to generate the initital median spline
                                    # fit to the data
###################################

######## data setup ###############
fitsfile = sm.file_input()          # User interactive function for selecting
                                    # FITS file
array = sm.open_FITS(fitsfile)      # function opens user selected fits file
                                    # and extracts the image data to the array
                                    #'array'.
###################################

##### mask generation values ######
thresh_val = input('Enter threshold counts value: ') # prompts user for the
                                    # 'threshold' counts value found previously
                                    # using the 'thresh_find' script
W = input('Enter pixel margin value: ') # prompts user for a number of pixels
                                    # to widen the masked regions by.
###################################


def make_mask_vmanual(array,thresh_val,W):  # main function of the script
    ###### execution timer #####       # records the time at start of function
    start_time = time.time()           # execution
    ############################

    print "Running, please wait..."

    MASK = np.zeros(array.shape)       # initialises an empty array of zeros
                                       # the same shape as the data array which
                                       # will hold the mask.

    for i in range(0,array.shape[1]):  # for loop to cycle through all pixel
                                       # columns in the data array, i.e. in the
                                       # vertical direction across the echelle
                                       # orders
```

```
    CUT = array[:,i]                # 1D array holds all of the pixel data
                                    # of current column

    # 'threshold line generation as used in thresh_find script
    S_MED = sm.spline_med(CUT,N)
    threshline = np.zeros(len(CUT))
    threshline[:] = S_MED[:] + thresh_val

    # for loop to increase the widths of the masked peaks by W pixels
    for j in range(0,array.shape[0]):
        if(CUT[j]>=threshline[j]):
            MASK[j-W:j+W+1,i] = 100



######## exectution timer #######
t = time.time() - start_time
print("--- %s seconds ---" % round(t,2))  # outputs execution time
#################################

MASK_FILE = fitsfile[:-5] +'_MASK.fits' # creates filename for FITS file to
                                    # hold the mask.

# Following code checks if an instance of this filename exists in the
# working directory and if so appends the suffix [X], where X increases
# until the filename is unique within the directory
re_no = 0
while (os.path.exists(MASK_FILE)):
    re_no+=1
    if (MASK_FILE[-6]==']'):
        MASK_FILE = MASK_FILE[:-8]
    else: MASK_FILE = MASK_FILE[:-5]
    MASK_FILE = MASK_FILE+'['+`re_no`+'].fits'
    print MASK_FILE
pyfits.writeto(MASK_FILE,MASK)    # unique MASK FITS file created in the
                                  # working directory
```

```
    # plotting image histogram of array as well as mask for comparison
    fig = plt.figure()
    plt.imshow(array,origin='lower')
    title = 'Image array Histogram'
    fig.suptitle(title,fontsize=16)
    plt.colorbar()

    fig = plt.figure()
    plt.imshow(MASK,origin='lower')
    title = 'Image array MASK'
    fig.suptitle(title,fontsize=16)
    plt.colorbar()

    plt.show()
    return MASK


make_mask_vmanual(array,thresh_val,W)     # main function run for given image
                                          # data, theshold counts value and pixel
                                          # margin value.
```

7.4 SKYBM_v1707.py

```
# Name:
#    SKYBM_v1707
#
# Purpose:
#    Script to perform the background noise subtracton on echelle spectrograph
#    FITS file image data.
#
# Operation:
#    This script uses the mask created using the script make_mask_vM1707 to
#    differentiate between the data and the background of the corresponding
#    echelle image data. The code iterates through each pixel column in the
#    image array, i.e. across echelle orders creates median binned data points
#    from each of the un-masked regions and then fits an interpolated model
#    line to these points using a Savitzky-Golay filter. Each modelled column
#    is stored in the array 'X_MODEL'. The same modeling procedure is then
#    applied to this model horizontally, resulting in a smooth model of the
```

```
#    shape of the background light contribution, 'Y_MODEL'. This model is then
#    subtracted from the image array to give the background subtracted image
#    array 'BGSUB'. The background subtracted image is then output to a FITS
#    file in the working directory.
#
#    The option exists to only use a user defined subsection of the image array
#    should only a certain spectral region be of interest. Should this option be
#    used the MASK image will be trimmed accordingly.
#
# Author:
#    Calum McCune, 2015 (Open University)



### SETUP ###
######## LIBRARIES #################
import matplotlib.pyplot as plt     # library for plotting functions
import numpy as np                   # library for array handling functions
import pyfits                        # library for handling fits files
import SKYBM_functions_v1707 as sm # library containing all functions
                                     # created specifically for this program.
import time                          # library for timing status checks
import os.path                       # library used to check the existence of
                                     # files
import scipy.signal as scisi        # library containing the sav-gol fitting
                                     # function
import sys                           # library containing function for exiting
                                     # script execution
####################################

######## SETUP ####################
yes = set(['yes','y'])              # 'yes' and 'no' sets for use in Y/N user
no = set(['no', 'n'])               # selections.
plot_int = 0                        # integer value for user control of plots
####################################


def SKYBM(array,MASK,FITS_file,plot_int): # function to model and subtract the
                                     # background from the image array.Function
```

```
                                # takes as input the image data array, the
                                # mask, the filename and the integer value
                                # controlling the plotting

X_MODEL = np.copy(array)        # creates a copy of the raw image array

for i in range(0,array.shape[1]):   # for loop to cycle through all pixel
                                # columns in the data array, i.e. in the
                                # vertical direction across the echelle
                                # orders. Produces a model line for each
                                # column to fill-in the 'X_MODEL' array

    # 1D arrays created holding all pixel data in column i of the image and
    # mask
    cut = array[:,i]
    mask_cut = MASK[:,i]

    # function sm.inter_points() examines the un-masked background regions
    # between the echelle orders and returns the central pixel locations and
    # local median data in the arrays mid_loc and med_data respectively.
    mid_loc = sm.inter_points(cut,mask_cut)[0]
    med_data = sm.inter_points(cut,mask_cut)[1]

    # adds 'ends' to the mid_loc array so the resulting interpolation line
    # can span the full length of the 'cut' array
    mid_loc = np.append(0,mid_loc)
    mid_loc = np.append(mid_loc,array.shape[0])
    win_len = len(med_data)/2    # window length 'win_len' required for
    if win_len % 2 == 0:         # savgol_filter must be a positive odd
        win_len +=1              # integer, if statement corrects if
                                 # necessary.

    sav_cut = scisi.savgol_filter(med_data,win_len,3) # savgol_fitler()
                                # function used to generate a smooth line
                                # interpolation of the med_data points.

    sav_cut = np.append(cut[0],sav_cut) # the first data point in sav_cut
                                # corresponds to the median value at the
                                # centre of the first un-masked region
```

```
                                # hence this model does not begin at pixel
                                # zero. Therefore the start point
                                # corresponding to pixel zero in the 'cut'
                                # array is added

    # since 'sav_cut' holds the smoothed interpolation of the 'med_data'
    # points it is only of the same size as 'med_data' and therefore needs
    # to be spread out it n order to model the full 'cut' array. To do this
    # an empty array 'SG_line' with size equal to 'cut' is created.
    # The values from sav_cut are then inserted at the pixel location given
    # by 'mid_loc', and a for loop uses 'straight line' array values to
    # connect the dots, filling in the model line for the full length of
    # the 'cut' array.
    SG_line = np.zeros(array.shape[0])
    SG_line[0] = cut[0]         # sets 'dot' to be joined to the sav_cut
                                # sets initial value in model line to equal
                                # to pixel zero of 'cut'.

    # for loop fills in the line completing the model
    for pt in range(1,len(sav_cut)):
        c = np.linspace(sav_cut[pt-1],sav_cut[pt],mid_loc[pt]-mid_loc[pt-1]+1)
        SG_line[mid_loc[pt-1]+1:mid_loc[pt]+1] = c[1:]

    # in the case where the final few pixels in the 'cut' are un-masked the
    # local median point will lie between the last masked region and the
    # last pixel. Therefore the model line 'SG_line' will not be filled-in
    # in this region. The following code fills in this region with the value
    # of the last smoothed data pixel.
    if SG_line[-1] == 0:
        b = 1
        while SG_line[-b] == 0:
            b += 1
        SG_line[-b+1:] = SG_line[-b]

    X_MODEL[:,i] = SG_line      # finished background model line for
                                # column i stored in coresponding column
                                # in array 'X_MODEL'

# After for loop above has run to completion the array 'X_MODEL' will now
```

```
# hold a smooth model fit line for each vertical column of the image array.
# This model will not therefore be smooth across horizontal rows.
Y_MODEL = np.copy(X_MODEL)          # 'Y_MODEL' array created to hold a copy
                                    # of 'X_MODEL' to be smoothed horizontally

# for loop uses savgol_filter() function to smooth all rows in the
# horizontal direction. Resulting in the finish background noise model
# 'Y_MODEL'
for i in range(0,X_MODEL.shape[0]):
    Y_sg =scisi.savgol_filter(X_MODEL[i,:],205,3)
    Y_MODEL[i,:] = Y_sg

### BACKGROUND SUBTRACTION!!! ###
BGSUB = array - Y_MODEL             # background noise model 'Y_MODEL' is now
                                    # subtracted from the image array creating
                                    # the background subtracted array 'BGSUB'
#################################

# plotting data: if the user chooses to display the raw and background
# subtracted image histograms, the value plot_int will have been set to 1.
# and the following code will be executed.
if plot_int == 1:
    fig = plt.figure()
    plt.imshow(array,origin='lower')
    title = ' image data Histogram'
    fig.suptitle(FITS_file+title,fontsize=16)
    plt.colorbar()
    fig = plt.figure()
    plt.imshow(BGSUB,origin='lower')
    title = ' \'background subttracted\' image Histogram'
    fig.suptitle(FITS_file+title,fontsize=16)
    plt.colorbar()
    plt.show()

return BGSUB                        # this function returns the background
                                    # subtracted image as the output
```

```
def main():          # main function of the script
    ### USER INTRODUCTION ###
    print 'SKY BACKGROUND MODEL PROGRAM'
    raw_input('\nPress any key to continue...\n\n') # awaits any user input
                # to continue with the program.

    ### FILE SELECTION & ARRAY CREATION ###
    FITS_file = sm.file_input()      # User interactive function for selecting
                                     # echelle spectra FITS file

    # following code extracts the image data from the FITS file
    hdu_list = pyfits.open(FITS_file)
    print '\n', FITS_file,' open\n'
    hdu = hdu_list[0]
    image = hdu.data

    # User prompted to choose if they want the resulting background subtracted
    # image to be displayed, with original for comparrison.
    choice = raw_input('Display raw and background subtracted image histograms?
(Y/N): ').lower()
    if choice in yes:
        plot_int = 1
    elif choice in no:
        plot_int = 0
    else: print 'Invalid selection!'


    # Function sm.array_size() allows for a user defined sub-section of the main
    # image array to be used. This is useful if only a small area of the CCD
    # chip is off interest and cuts down on execution time.
    # The function outputs the users chosen dimensions which can then be used to
    # identically trim the mask.
    array,x_min,x_max,y_min,y_max = sm.array_size(image)


    MASK = sm.mask_open()            # function opens the corresponding mask FITS
                                     # file and store image data in the array
                                     # 'MASK'
```

```
    ###### execution timer #####        # records the time at start of function
    start_time = time.time()            # execution
    ############################

    # following code used to deal with the scenario where the mask and the image
    # shapes do not match. In the case that a smaller region of the image has
    # been designated by the user then this code will correct the mask
    # accordingly using the dimensions given in array_size(). Else an incorrect
    # mask has been used and the user is prompted to restart the script with the
    # appropriate mask
    if MASK.shape != array.shape:
        print '\nWARNING: MASK and array shapes are different!'
        if (MASK.shape > array.shape):
            print '\nMASK is larger than array!'
            choice = raw_input('\nUse same sub section of MASK as array?
(y/n)').lower()
            if choice in yes:
                MASK = MASK[x_min:x_max,y_min:y_max]
            elif choice in no:
                sys.stdout.write('\nPlease restart with appropiate MASK')
        elif (MASK.shape < array.shape):
            print '\nMASK is smaller than array!'
            print '\nPlease enter array region matching this MASK;'
            X_min = input('x min: ')
            X_max = input('x max: ')
            Y_min = input('y min: ')
            Y_max = input('y max: ')
            sub = image[X_min:X_max,Y_min:Y_max]
            array = sub

    BGSUB = SKYBM(array,MASK,FITS_file,plot_int) # SKYBM() function is
                                    # implemented to perform the background
                                    # subtraction resulting in the array 'BGSUB'


    BG_FILE = 'BG_'+FITS_file       # new filename created for the output
                                    # fitsfile that will contain the resultant
                                    # background subtracted image
```

```
    # Following code checks if an instance of this filename exists in the
    # working directory and if so appends the suffix [X], where X increases
    # until the filename is unique within the directory
    re_no = 0
    while (os.path.exists(BG_FILE)):
        re_no+=1
        if (BG_FILE[-6]==']'):
            BG_FILE = BG_FILE[:-8]
        else: BG_FILE = BG_FILE[:-5]
        BG_FILE = BG_FILE+'['+`re_no`+'].fits'
        print BG_FILE

    pyfits.writeto(BG_FILE,BGSUB)     # unique BGSUB FITS file created in the
                                      # working directory

    ######## exectution timer #######
    t = time.time() - start_time
    print("--- %s seconds ---" % round(t,2)) # outputs execution time
    ################################

    hdu_list.close()                  # closes the original FITS file

main()                                # main function is run
```

Appendix 8: Research Preparation

Coming from a mostly physics background I had very little exposure to Astrophysics, my only experience coming from the subject being touched upon in A-levels and again in a single module in my under-graduate degree. It was therefore necessary to read and familiarise myself with the specific mathematics, methodology and terminology of the field prior to commencing my studies. Here I give an account of the preparation and work I have done so far over the summer and in the first weeks of my studies.

Summer preparation

An Introduction to Astrophysics – A. Norton

I read the second chapter of "An Introduction to Astrophysics – A. Norton" which covered the basics of the subject. I learnt that there are three main parameters by which we can determine the behaviour of stars, namely their mass, age and composition. Each of which can be inferred from just measuring the light (EM radiation, inclusive) these stars emit. From these basic properties it is possible to determine a whole host of the stars other measurable characteristics, such as their temperature, luminosity and radii.

This chapter also taught me the convention for naming stars as per their location within a known constellation or from a catalogue/survey. It also covered how to refer to the positions of celestial bodies in terms of "terrestrial latitude & longitude", Declination and Right Ascension respectively. Given that from our perspective the sky appears as a flat sphere it is useful to refer to apparent separation of celestial objects in terms of the angular area they occupy in terms of degrees, arcmin (1/60 degrees) and arcsec (1/3600 degrees).

The chapter familiarised me with the units used in astrophysics, as well as the conversions for when non-SI units are being used. Since for the most part the scale of units we use for

everyday terrestrial measurements in so minuscule compared with the astronomical scale, it is convenient to express astrophysical quantities in units relative to the sun. As such measurements of astronomical bodies are often given in terms of M_{sol}, L_{sol} and R_{sol}.

Distances too are orders of magnitude greater than anything we experience on Earth and so alternative units are used to make the numbers more manageable. Distances are either given in relation to the average distance between the Earth and the sun (Astronomical Units, AU), in relation to the degree of observational parallax (parsec, pc), or in terms of the speed of light (lightyears, ly).

As we leave the terrestrial scale for those of the astronomical the velocities become enormous. However this is not immediately apparent given the extreme distances and relative day to day static appearance of the night's sky, velocities of even hundreds of kilometres per second would result in an extremely small angular shift over the course of a year. From an observational perspective it is convenient to split the overall motion into transverse and radial, since we can only see the transverse component directly and the radial must be inferred by other means. The relationship between spectra and temperature was established. How spectral lines indicated the emitting compositional species which in turn indicated the temperature and nature of internal reactions. This chapter gave me a broad introduction to the many terms, units, techniques and mathematics of Astrophysics, which was essential for my continued reading on the subject.

Other topics covered include:

- Fluxes and Luminosities
- The Doppler effect and red-shift
- Astronomical magnitudes (apparent and absolute)
- The HR diagram and how it describes stellar evolution and population

- Measuring the masses of stars through companion interactions (such as planets, binaries or comets)
- Radial velocity curves
- Nuclear processes and the lifecycles and eventual deaths of stars.

Transiting Exoplanets – C. Haswell

Following the previous books introduction to the basics of astrophysics I then read and worked through the textbook "Transiting Exoplanets – C. Haswell" to get a more detailed knowledge around my subject matter of exoplanet observations.

The book first considers making observations of our solar system as if it were that of a distant star and considers the ease of detection of the planets from an outside observer perspective via various detection methods. We find out that even though we know of nine planets to inhabit our system, this would not be so apparent from the perspective of a distant extra-terrestrial astronomer with our current technology.

The main focus of this book was on exoplanet observations via the transit method. When a planet passes between its host star and an observer it blocks some of the light from the stellar disc. This results in a fractional decrease in the light coming from a star producing a characteristic transit light curve from which we can study the properties of the planet.

The limitations and advantages of the transit method were discussed. By observing transits the orbital inclination, i, can be inferred which alludes other detection methods. However this method strongly favours close in giant planets around dim host stars limiting the potential candidates for detection. Additionally only planets whose orbits fall across the disk of their host star i.e. edge on from our perspective, can be seen to transit. These two limitations can be shown in the expression for the geometric transit probability:

$$geometric\ tranist\ probability = \frac{R_* + R_P}{a} \approx \frac{R_*}{a}$$

Where R_* and R_P are the radii of the star and planet respectively and a is the orbital distance of the planet.

The transit light curve is produced by plotting the flux over time for an observed star. This flux remains roughly constant until the planet transits producing a characteristic dip. From looking at key features of the shape of the transit light curve we are able to infer such properties as radius, impact parameter, orbital inclination and even spectrographic details of the exoplanet atmosphere! If an observed planet has an atmosphere, and the transit light curve is of good quality, it is possible to infer atmospheric composition from the absorption lines caused in the stellar spectrum.

The current (c.2010) state of the known exoplanet population was covered in terms of their respective methods of first detection. Every detection method has its own characteristic advantages and limitations, this means that there are biases for their relative success in terms of planetary discoveries. As mentioned before the transit method favours giant planets in close orbits around faint stars whereas, for example, the direct imaging method favours planets in wide orbits. These biases are represented in the demographic of discovered planets. At the time the proportion of planet s discovered via the transit method was ~40% (by the end of 2010) becoming the most successful method by 2014 at ~68% (808 from Kepler in just this year).

The concept of equilibrium temperature of a planet was discussed. With measurements of the proximity to, and flux from, a host star as well as the reflective albedo of the planet allows for surface temperature to be calculated (assuming no thermal effects such as radiative transfer or atmospheric insolation).

When a star has a transiting planet (or eclipsing binary) it is possible to determine that stars rotation speed, direction and orientation due to a phenomenon called the Rossiter-McLaughlin effect. As the host star rotates on its axis one quadrant will be moving towards the observer and the opposite quadrant will be moving away, resulting in the stellar spectrum being blue-shifted and red-shifted respectively. As the planet passes across the stellar disc it will block out sections of the stars surface giving rise to a time-varying Doppler shift in the transit light curve. Examining this Doppler shift we can determine the stars rotational speed, direction and the orientation of its axis. Comparison of these rotational characteristics with the planets orbital inclination and direction can be used to investigate migrations and past interactions within the system.

Secondary Eclipses and phase variations can occur as a planet begins to pass behind its host star at superior conjunction. This can result in an increase in the detected flux from the star as a result of starlight being reflected from the planet's surface towards the observer. Such an effect would be additional confirmation of the planets presence as it would offer another predictable time event observable in the light curve.

Transit timing variations (TTV) and orbital dynamics was discussed. TTVs occur when gravitational interactions cause changes in the orbital dynamics of the system resulting in changing the precise timing of transits. These effects can indicate the presence of additional undetected orbiting bodies in the system such additional planets, exotrojans (bodies occupying Lagrangian points) or even exomoons. This represents a way in which transit observations can result in indirect detection of planets around a star other than the transiting planet.

I found this text incredibly helpful with familiarising myself with the methodology and concepts of exoplanet observations and specifically by the transit method. In addition I learnt about what information it is possible to infer from observations of exo-planetary systems and how this information can be applied to learning about our own solar systems formation and evolution.

Python tutorials

Anticipating the need for a strong programming platform to analyse data Python was a good choice. Renowned for being user friendly, it is also widely adopted and free. Python came recommended by most people I asked on the matter. Without knowing the specifics of what coding for astrophysics would involve I got to work familiarising myself with the basic commands and code, i.e. for loop, if statements maths operators etc. This was done using a collection of online video tutorials.

Exoplanets Summer School

Summer School (Workshops, Labs, Field, Verdict)

In further preparation for my studies there was the fantastic opportunity of a dedicated exoplanet summer school hosted on the island of La Palma, Canary Islands, which happens to be the site of numerous telescope groups including SuperWASP, GTC (Gran Telescopio de Canarias) and most notably NOT (Nordic Optical Telescope) from which we would be observing and collecting real data.

Each of the nine days was packed full of activities with the overall division of time being roughly 40% lectures and discussion, 30% field trips (tour of the lava fields and telescope facilities), 15% Lab Group exercises and the remaining 15% being free-time. A more detailed look at the content of the course is given below.

Poster Presentation

Select course participants gave 4 minute pitch-presentations for their respective posters to be shown at the later poster presentation session. Although relevant to exoplanets nearly all posters were of unique subject backgrounds, for example tidal-heating and habitability of exo-moons, microlensing, evaporating rocky planets, atmospheric modelling, observational data of exotic planetary systems, potential future detection methods, some future telescope projects and many more. This part of the course went to show that although a, relatively, new emerging field of study there are multiple lines of inquiry under the umbrella of exo-planet research. The poster presentations happened over two nights and gave each of the students who had brought a poster with them the chance to pitch their poster to generate some interest before the poster session where interested participants could find them by their posters for a more detailed breakdown of their research.

Poster sessions

After the presentations all participants had the opportunity to walk around the posters where the authors would be available to give a more detailed presentation and answer questions. This provided a much more personal side to the poster segment as questions could be asked one-on-one without the 4 minute time limit. This session was much more relaxed and served as a good ice-breaker at the beginning of the course. Some students including myself did not provide posters. My own reasoning being that the only subjects/work that I could competently present would not have been at all relevant to the subject matter. Although having experienced it as purely an observer I would be interested in having presentable work for the next such occasion, if nothing else it would serve as a great talking point and an

opportunity to familiarise your international peers with your work. Perhaps it could be useful for networking.

Lectures

The lectures were given by visiting lecturers from various universities, projects and missions and covered a wide range of topics relevant to Exoplanets. Each was 45 minutes long followed by 15 minutes of questions and discussion.

Lectures included: Physics and structure of proto-planetary disks, Early Evolution of the solar system, Formation of rocky planets, Types of Exoplanets, Methods for detection of exoplanets, Search for exoplanets around solar-type stars, Highlights from the Kepler mission, The PLATO mission, Observation of spectra of exoplanets, Detection of exomoons, Habitable satellites, Characterisation of exoplanet atmospheres, Climate and (exo)planet atmospheres, Habitability around different types of stars, Atmospheric Biosignatures, Water delivery on planets and Comets in the solar and extrasolar systems.

These lectures were a great way of exploring all the many different avenues of exoplanet research and the current state of understanding as well as familiarising me with the terminology and concepts. It was also an opportunity to ask questions of experts in these differing fields.

Although the quality of the lecturers was very good there were a few who really stood out who were very talented and describing their topics and keeping the audience interest. This was notable as it offered an insight into how best to present information and communicate ideas, the take away message less equations/numbers/paragraphs, more imagery, method pathway, pictures, presentable and significant data and where possible humour.

Group Exercises

Preparation Lectures: In addition to the standard lectures there were a few preparation lectures. Unlike the other lectures these talks were aimed at preparing participants for the group exercise sessions. These lectures were on the methodology of modelling of exoplanet atmospheres, the details and workings of the Nordic Optical Telescope and an introduction to the observation project.

Observation Project: Working with the NOT, a known transit candidate was selected and the telescope trained on it for the duration of the transit. For our night our candidate was GJ 1214 b. With the help of the telescope operators the appropriate settings were selected for the observation, such as wavelength range and exposure time. The telescope was then set-up and observed the star for the duration of the transit with a non-transit time margin on either side. The data was then transferred to the students whose laptops had been setup with the appropriate software for the subsequent data analysis session. This gave me a great opportunity to see how observational data for this area of research is obtained as well as experiencing first-hand the problem that can arise when we had to close down the telescope prematurely due to rising humidity that could have led to moisture build up on the mirror.

Data Reduction: Most groups were fortunate enough to have the data reduction session come on the day after their observation night so they could work with their actually data, for those that did not Kepler data was provided. Fortunately as the first group observing we got to work with our data. It was suggested that we use the software package IRAF for our data reductions however it was left up to the individual's preference. In my group it was decided to split up and take three separate programming approaches and reconvene to compare the data. Being new to this form of data reduction this provided me with a fantastic opportunity to see how these different packages dealt with the task. Having no prior knowledge of any of the chosen languages I was allowed to watch the data reduction process taking notes on

exactly what was being done to the data and why. Although I did not understand the actual code that was being written I felt I followed the steps and I was very excited to see the actual transit curve that came out of the data.

Modelling of Exoplanet Atmospheres: For this session we had been asked previously, if we had a Mac or a Linux laptop, to download a suite of Python tools and datasets that would allow us to run our models. Not having a Mac or Linux PC of my own one was provided for me courtesy of the University of Stockholm. Using the designated software we performed several exercises modelling atmospheric characteristics of different planets. The exercises looked at the chemical interactions giving rise to ozone, producing 1D temperature profiles and atmospheric distributions of a Jupiter-like planet, and the temperature profile on a planet when placed around host stars of different spectral class. This session gave me an insight into the operation of Linux computers as well as my first experience of using Python to handle data. It was a useful insight into the construction and implementation of models for analysing data.

Student-led discussions

The student led discussion sessions served as a chance for the participants of the course to discuss certain topics which had been covered. Without the input from the lecturers the idea was that as a group the students could talk freely about the topics they had little experience in and to offer thoughts, make contributions and insights.

Exoplanet Atmospheres: There had been a lot of content on exoplanet atmospheres over the course. This session gave students the opportunity to discuss what had been learned. This discussion was actually led by a visiting professor. This did mean that detailed questions could be asked but this also led to a reserved atmosphere where little contribution was actually made from the audience.

Biomarkers and the search for life on exo-planets: This session was a lot more speculative. A lot less had been covered on this topic throughout the course and certainly no hard data. This time a group of students had taken it upon themselves to take over the organisation, structure and presentation of the discussion. It was decided to first encourage out of the box thinking, which is necessary for this topic matter. It was decided to break the audience up into the groups they had been working in all week, the thought being that this way each group member's contributions would be heard. The discussion was broken into several sections pertaining to the question of extra-terrestrial life, namely habitability, biomarkers and instrumentation. For each section all groups had 10 minutes to discuss and then one member was nominated to present the groups ideas to the main audience. Since this discussion was largely speculation, however scientifically- guided, and no one was an expert, this was one of the moments in the course where I managed to make a real contribution. This was one of my favourite parts of the course as it was a real meeting of minds from different disciplines, whenever someone's thoughts led them into a place of their own ignorance there was inevitably someone in the group who was more knowledgeable and able to guide that train of thought either further on or back to reality.

Post summer reading & development

Since starting at the OU I have been using my time to further develop my knowledge and skills which will be relevant to my research. To familiarise myself further with astronomical spectroscopy in general I have started reading Astronomical Spectroscopy – Tennyson, a book which comes recommended by my fellow post-grad students. Nearly all information about the universe comes from the study of light, therefore understanding what information it contains and how it is extracted will always prove useful in my future astronomy work.

Following my introduction to Python over the summer and witnessing how it can be implicated for astronomy I have been further working on my Python skills. Building on the basics I learned over the summer I am now focussing on the mathematical and science tools useful to astrophysical calculations. I have also been reading scientific papers relevant to exoplanet research. This has been useful for gaining a better understanding of the methodology and terminology used, as well as providing me with more background information on the development of the field.

I will be beginning my research by conducting data analysis of Échelle spectrographs from the SOPHIE (Spectrographe pour l'Observation des Phénomènes des Intérieurs stellaires et des Exoplanètes) telescope. This will be done using the software package ECHOMOP which operates on Linux. Prior to having access to the Linux server I had been reading up on the specifics of Échelle spectroscopy as well as the operation and tools of the ECHOMOP platform. Since getting access to the Linux server I have been practising using the ECHOMOP package on data from two test stars. Following the process kindly explained to me by John Barnes and with reference to the extensive ECHOMOP manual I was able to perform data reductions and produce Échelle spectrographs for both stars.